# Learning Languages from Positive Data and Negative Counterexamples

Sanjay Jain [a,1] Efim Kinber [b]

[a] *School of Computing, National University of Singapore, Singapore 117543. Email: sanjay@comp.nus.edu.sg*

[b] *Department of Computer Science, Sacred Heart University, Fairfield, CT 06432-1000, U.S.A. Email: kinbere@sacredheart.edu*

**Abstract**

In this paper we introduce a paradigm for learning in the limit of potentially infinite languages from all positive data and negative counterexamples provided in response to the conjectures made by the learner. Several variants of this paradigm are considered that reflect different conditions/constraints on the type and size of negative counterexamples and on the time for obtaining them. In particular, we consider the models where 1) a learner gets the least negative counterexample; 2) the size of a negative counterexample must be bounded by the size of the positive data seen so far; 3) a counterexample can be delayed. Learning power, limitations of these models, relationships between them, as well as their relationships with classical paradigms for learning languages in the limit (without negative counterexamples) are explored. Several surprising results are obtained. In particular, for Gold's model of learning requiring a learner to syntactically stabilize on correct conjectures, learners getting negative counterexamples immediately turn out to be as powerful as the ones that do not get them for indefinitely (but finitely) long time (or are only told that their latest conjecture is not a subset of the target language, without any specific negative counterexample). Another result shows that for behaviourally correct learning (where semantic convergence is required from a learner) with negative counterexamples, a learner making just one error in almost all its conjectures has the "ultimate power": it can learn the class of all recursively enumerable languages. Yet another result demonstrates that sometimes positive data and negative counterexamples provided by a teacher are not enough to compensate for full positive and negative data.

[1] Supported in part by NUS grant number R252-000-127-112.

## 1 Introduction

Defining a computational model adequately describing learning languages is an important long-standing problem. In his classical paper [Gol67], M. Gold introduced two major computational models for learning languages. One of them, learning from texts, assumes that the learner receives all positive language data, i.e., all correct statements of the language. The other model, learning from informants, assumes that the learner receives all correct statements of the languages, as well as all other (incorrect) statements, appropriately labeled as incorrect, that can be potentially formed within the given alphabet. In both cases, a successful learner stabilizes on a correct description of the target language, i.e., a grammar for the target language. J. Barzdin [Bār74] and J. Case and C. Smith [CS83] introduced a different, more powerful model called *behaviorally correct* learning. A behaviorally correct learner almost always outputs conjectures (not necessarily the same) correctly describing the target language. An important feature of all these models is that they describe a process of learning in the limit: the learner stabilizes on the correct conjecture (or conjectures), but does not know when it happens. The above seminal models, doubtless, represent certain important aspects of the process of learning potentially infinite targets. On the other hand, when we consider how a child learns a language communicating with a teacher, it becomes clear that these models reflect two extremes of this process: positive data only is certainly less than what a child actually gets in the learning process, while informant (the characteristic function of the language) is much more than what a learner can expect (see for example, [BH70,HPTS84,DPS86]).

D. Angluin, in another seminal paper [Ang88], introduced a different important learning paradigm, i.e., learning from queries to a teacher (oracle). This model, explored in different contexts, including learning languages (see, for example, [LNZ02,LZ04b,LZ04a]), addresses a very important tool available to a child (or any other reasonable learner), i.e., queries to a teacher. However, in the context of learning languages, this model does not adequately reflect the fact that a learner, in the long process of acquisition of a new language, potentially gets access to all correct statements. (Exploration of computability via queries to oracles has a long tradition in the theory of computation in general [Rog67,GM98], as well as in the context of learning in the limit [GP89,FGJ$^+$94,LNZ02]. Whereas in most cases answers to queries are sometimes not algorithmically answerable - which is the case in our model, or computationally NP or even harder - as in [Ang88], exploring computability or learnability via oracles often provides a deeper insight on the nature and capabilities of both).

In this paper, we combine learning from positive data and learning from queries into a computational model, where a learner gets all positive data and can ask a teacher if a current conjecture (a grammar) does not generate wrong statements (questions of this kind can be formalized as *subset queries*, cf. [Ang88]). If the conjecture does generate a wrong statement, then the teacher gives an example of such a statement (a negative counterexample) to the learner. In our main model, we assume that the teacher immediately provides a negative counterexample if it exists. However, in many situations, a teacher may obviously need a lot of time to determine if the current conjecture generates incorrect statements. [2] Therefore, we consider two more variants of our main model that reflect this problem. In the first variant, the teacher is not able to provide a negative counterexample unless there is one whose size does not exceed the size of the longest statement seen so far by the learner. In the second variant, the teacher may delay providing a negative counterexample (and, eventually, may even simply answer that the conjecture is excessive, i.e., without providing any negative counterexamples!). Interestingly, while the former model is shown to be weaker than the main model, the latter one turns out to be as powerful as the main model (in terms of capabilities of a learner; we do not discuss related complexity issues – such as how providing counterexamples quickly may speed up convergence to a right conjecture)!

Our goal in this paper is to explore the new models of learning languages, their relationships, and how they fair in comparison with other popular learning paradigms. In particular, we explore how quality and availability of the negative counterexamples to the conjectures affects learnability. Note that learning from positive data and a finite amount of negative information was explored in [BCJ95]. However, unlike arbitrary negative counterexamples in our model, negative data in [BCJ95] is *preselected* to ensure that just a small number of negative examples (or, just one example) can greatly enhance capabilities of a learner. Shinohara [Shi86] considered a model of negative data, where *any* $n$ negative examples might be given to the learner. Motoki [Mot91] considered supplying negative data to the learner, which contains a preselected subset of the complement of the input (this is similar to one of the models considered in [BCJ95]).

The paper is structured as follows. In Section 2 we introduce necessary notation and basic definitions needed for the rest of the paper. In particular, we define some variants of the classical Gold's model of learning from texts

[2] Note that an oracle-teacher in our model must possess knowledge of the complete characteristic function of the target language. It is hard to imagine that an individual real-world teacher can fully possess such knowledge. Therefore, by "teacher" in our discussion one should imagine a sort of a "linguistic community" with the knowledge of the characteristic function in question distributed between its members, rather than just an individual teacher. However, from the standpoint of mathematical feasibility, immediate implementation of this understanding of a "teacher" would most likely be rather awkward, therefore we have given our preference to the classical model of an oracle.

(positive data) and informants (both positive and negative data), **TxtEx** and **InfEx**, as well as its behaviorally correct counterpart **TxtBc** and **InfBc**.

In Section 3 we define our four models for learning languages from texts and negative counterexamples. In the first, basic, model, a learner is provided a negative counterexample every time it outputs a hypothesis containing elements not belonging to the target language. The second model is a variant of the basic model when a learner receives the least negative counterexample. The third model takes into account some complexity constraints. Namely, the learner receives a negative counterexample only if there exists one whose size is bounded by the size of the longest positive example seen in the input so far. The fourth model slightly relaxes the constraint of the model three: the size of the negative counterexample must be bounded by the value of some function applied to the size of the longest positive example in the input. We also introduce non-recursive variants of all four models - when the learner is not necessarily computable.

Section 4 is devoted to **Ex**-style learning from positive data and negative counterexamples: the learner eventually stabilizes on a correct grammar for the target language. First, in order to demonstrate the power of our basic model, we show that any indexed class of recursively enumerable languages can be learned by a suitable learner in this model. Then we show that the second model is equivalent to the basic model: providing the least negative counterexample does not enhance the power of a learner. In our next major result (Theorem 19) we show that there is a class of languages learnable from informants and not learnable in our basic model. This means that sometimes negative counterexamples are not enough - the learner must have access to all statements not belonging to the language! (This result follows from a more general result for **Bc**-style learning proved in Section 6). In particular, this result establishes certain constraints on the learning power of our basic model. The proof of this result employs a new diagonalization technique working against machines learning via negative counterexamples. We also establish a hierarchy of learning capabilities in our basic model based on the number of errors that learner is allowed to have in the final hypothesis. Then we consider the two models with restricted size of negative counterexamples (described above). We show that these models are different from and weaker than our basic model. Still we show that these models are quite powerful: firstly, if restricted to the classes of infinite languages, they are equivalent to the basic model, and, secondly there are learnable classes in these models that cannot be learned in classical **Bc**-model (without negative counterexamples) - even if an arbitrary finite number of errors is allowed in the final conjectures. In the end of the section we demonstrate that a non-recursive learner in our basic model can learn the class of all recursively enumerable languages. In fact, non-recursive learning with negative counterexamples turns out to be equivalent to non-recursive learning from informants (in contrast to Theorem 19 for computable learners mentioned above).

In Section 5 we introduce the concept of a *locking sequence* similar to the one defined in [BB75] for the classical **Ex**-style learning. As in the case of the classical **Ex**-model, locking sequences turn out to be useful in characterizing learnability within our model. In particular, locking sequences are employed in our next result presented in this section. This result (Theorem 43) demonstrates that models of learning from positive data and negative counterexample where the teacher may delay providing a negative counterexample (we define four natural versions of them) are still equivalent to the basic model with no delays!

Section 6 is devoted to our **Bc**-style models. As in the case of **Ex**-style learning, we show that providing the least negative counterexample does not enhance the power of a learner. We also show that learning with restricted size of negative counterexamples is weaker than the basic model in this setting. In particular, we show that there exists an indexed class of recursively enumerable languages that cannot be learned with negative counterexamples of restricted size (note that all such classes are learnable in our basic **Ex**-style model as stated in Theorem 14). Then we show that languages learnable in our basic **Bc**-style model (without errors) are **Bc**-learnable from informants. In the end we establish one of our most surprising results. First, we demonstrate that the power of our basic **Bc**-style model is limited: there are classes of recursively enumerable languages not learnable in this model if no errors in almost all (final) conjectures are allowed. On the other hand, there exists a learner that can learn all recursively enumerable languages in this model with at most one error in almost all correct conjectures! (The learner here needs to find answers to undecidable questions concerning comparison of target and hypothesis languages; the teacher cannot always provide negative counterexamples to the languages different from the target (for example when the conjecture is a subset of the target language), however, by possibly making just one deliberate error, the learner finds a way to encode its questions into conjectures so that the teacher is forced to provide negative counterexamples giving out the necessary information). Based on similar ideas, we obtain some other related results - in particular, that, with one error allowed in almost all correct conjectures, the class of all infinite recursively enumerable languages is learnable in the model with restricted size of negative counterexamples. In contrast to the case with no errors, we also show that when errors are allowed, **Bc**-learning from informants is a proper subset of our basic model of **Bc**-learning with errors and negative counterexamples (Corollary 54).

## 2 Notation and Preliminaries

Any unexplained recursion theoretic notation is from [Rog67]. The symbol $N$ denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$. Symbols $\emptyset$, $\subseteq$, $\subsetneq$, $\supseteq$, and $\supsetneq$ denote empty set, subset, proper subset, superset, and proper superset, respectively. Cardinality of a set $S$ is denoted by $\mathrm{card}(S)$. The maximum and minimum of a set are denoted by $\max(\cdot), \min(\cdot)$, respectively, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. $L_1 \boldsymbol{\Delta} L_2$ denotes the symmetric difference of $L_1$ and $L_2$, that is $L_1 \boldsymbol{\Delta} L_2 = (L_1 - L_2) \cup (L_2 - L_1)$. For a natural number $a$, we say that $L_1 =^a L_2$, iff $\mathrm{card}(L_1 \boldsymbol{\Delta} L_2) \leq a$. We say that $L_1 =^* L_2$, iff $\mathrm{card}(L_1 \boldsymbol{\Delta} L_2) < \infty$. Thus, we take $n < * < \infty$, for all $n \in N$. If $L_1 =^a L_2$, then we say that $L_1$ is an $a$-variant of $L_2$.

We let $\langle \cdot, \cdot \rangle$ stand for an arbitrary, computable, bijective mapping from $N \times N$ onto $N$ [Rog67]. We assume without loss of generality that $\langle \cdot, \cdot \rangle$ is monotonically increasing in both of its arguments. We define $\pi_1(\langle x, y \rangle) = x$ and $\pi_2(\langle x, y \rangle) = y$.

By $\varphi$ we denote a fixed *acceptable* programming system for the partial computable functions mapping $N$ to $N$ [Rog67,MY78]. By $\varphi_i$ we denote the partial computable function computed by the program with number $i$ in the $\varphi$-system. Symbol $\mathcal{R}$ denotes the set of all recursive functions, that is total computable functions. By $\Phi$ we denote an arbitrary fixed Blum complexity measure [Blu67,HU79] for the $\varphi$-system. A partial recursive function $\Phi(\cdot, \cdot)$ is said to be a Blum complexity measure for $\varphi$, iff the following two conditions are satisfied:

(a) for all $i$ and $x$, $\Phi(i, x)\downarrow$ iff $\varphi_i(x)\downarrow$.

(b) the predicate: $P(i, x, t) \equiv \Phi(i, x) \leq t$ is decidable.

By convention we use $\Phi_i$ to denote the partial recursive function $\lambda x.\Phi(i, x)$. Intuitively, $\Phi_i(x)$ may be thought as the number of steps it takes to compute $\varphi_i(x)$.

By $W_i$ we denote $\mathrm{domain}(\varphi_i)$. $W_i$ is, then, the recursively enumerable (r.e.) set/language ($\subseteq N$) accepted (or equivalently, generated) by the $\varphi$-program $i$. We also say that $i$ is a grammar for $W_i$. Symbol $\mathcal{E}$ will denote the set of all r.e. languages. Symbol $L$, with or without decorations, ranges over $\mathcal{E}$. By $\chi_L$ we denote the characteristic function of $L$. By $\overline{L}$, we denote the complement of $L$, that is $N - L$. Symbol $\mathcal{L}$, with or without decorations, ranges over subsets of $\mathcal{E}$. By $W_{i,s}$ we denote the set $\{x < s \mid \Phi_i(x) < s\}$.

We now present concepts from language learning theory. The next definition introduces the concept of a *sequence* of data.

**Definition 1** (a) A *sequence* $\sigma$ is a mapping from an initial segment of $N$ into $(N \cup \{\#\})$. The empty sequence is denoted by $\Lambda$.

(b) The *content* of a sequence $\sigma$, denoted content($\sigma$), is the set of natural numbers in the range of $\sigma$.

(c) The *length* of $\sigma$, denoted by $|\sigma|$, is the number of elements in $\sigma$. So, $|\Lambda| = 0$.

(d) For $n \leq |\sigma|$, the initial sequence of $\sigma$ of length $n$ is denoted by $\sigma[n]$. So, $\sigma[0]$ is $\Lambda$.

Intuitively, #'s represent pauses in the presentation of data. We let $\sigma$, $\tau$, and $\gamma$, with or without decorations, range over finite sequences. We denote the sequence formed by the concatenation of $\tau$ at the end of $\sigma$ by $\sigma\tau$. Sometimes we abuse the notation and use $\sigma x$ to denote the concatenation of sequence $\sigma$ and the sequence of length 1 which contains the element $x$. SEQ denotes the set of all finite sequences.

**Definition 2** [Gol67] (a) A *text* $T$ for a language $L$ is a mapping from $N$ into $(N \cup \{\#\})$ such that $L$ is the set of natural numbers in the range of $T$. $T(i)$ represents the $(i+1)$-th element in the text.

(b) The *content* of a text $T$, denoted by content($T$), is the set of natural numbers in the range of $T$; that is, the language which $T$ is a text for.

(c) $T[n]$ denotes the finite initial sequence of $T$ with length $n$.

**Definition 3** A *language learning machine from texts* [Gol67] is an algorithmic device which computes a mapping from SEQ into $N$.

**Definition 4** *We say that a recursive function $I$ is* an informant *for $L$ iff for all $x$, $I(x) = \chi_L(x)$.*

Intuitively, informants give both all positive and all negative data for the language being learned. $I[n]$ is the first $n$ elements of the informant $I$. One can similarly define language learning machines from informants.

We let $\mathbf{M}$, with or without decorations, range over learning machines. $\mathbf{M}(T[n])$ (or $\mathbf{M}(I[n])$) is interpreted as the grammar (index for an accepting program) conjectured by the learning machine $\mathbf{M}$ on the initial sequence $T[n]$ (or $I[n]$). We say that $\mathbf{M}$ converges on $T$ to $i$, (written: $\mathbf{M}(T)\downarrow = i$) iff $(\forall^\infty n)[\mathbf{M}(T[n]) = i]$. Convergence on informants is similarly defined.

There are several criteria for a learning machine to be successful on a language. Below we define some of them. All of the criteria defined below are variants of the **Ex**-style and **Bc**-style learning described in the Introduction; in addition, they allow a finite number of errors in almost all conjectures (uniformly bounded, or arbitrary).

**Definition 5** [Gol67,CL82] Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{TxtEx}^a$*-identifies a text* $T$ just in case $(\exists i \mid W_i =^a \mathrm{content}(T))$ $(\forall^\infty n)[\mathbf{M}(T[n]) = i]$.

(b) $\mathbf{M}$ $\mathbf{TxtEx}^a$*-identifies an r.e. language* $L$ (written: $L \in \mathbf{TxtEx}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies each text for $L$.

(c) $\mathbf{M}$ $\mathbf{TxtEx}^a$*-identifies a class* $\mathcal{L}$ *of r.e. languages* (written: $\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies each language from $\mathcal{L}$.

(d) $\mathbf{TxtEx}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})]\}$.

**Definition 6** [CL82] Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{TxtBc}^a$*-identifies a text* $T$ just in case $(\forall^\infty n)[W_{\mathbf{M}(T[n])} =^a L]$.

(b) $\mathbf{M}$ $\mathbf{TxtBc}^a$*-identifies an r.e. language* $L$ (written: $L \in \mathbf{TxtBc}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtBc}^a$-identifies each text for $L$.

(c) $\mathbf{M}$ $\mathbf{TxtBc}^a$*-identifies a class* $\mathcal{L}$ *of r.e. languages* (written: $\mathcal{L} \subseteq \mathbf{TxtBc}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtBc}^a$-identifies each language from $\mathcal{L}$.

(d) $\mathbf{TxtBc}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtBc}^a(\mathbf{M})]\}$.

**Definition 7** [Gol67,CL82] Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{InfEx}^a$*-identifies* $L$ (written: $L \in \mathbf{InfEx}^a(L)$), just in case for informant $I$ for $L$, $(\exists i \mid W_i =^a L)$ $(\forall^\infty n)[\mathbf{M}(I[n]) = i]$.

(b) $\mathbf{M}$ $\mathbf{InfEx}^a$*-identifies a class* $\mathcal{L}$ *of r.e. languages* (written: $\mathcal{L} \subseteq \mathbf{InfEx}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{InfEx}^a$-identifies each language from $\mathcal{L}$.

(c) $\mathbf{InfEx}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{InfEx}^a(\mathbf{M})]\}$.

**Definition 8** [CL82] Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{InfBc}^a$*-identifies* $L$ (written: $L \in \mathbf{InfBc}^a(L)$), just in case for informant $I$ for $L$, $(\forall^\infty n)[W_{\mathbf{M}(I[n])} =^a L]$.

(b) $\mathbf{M}$ $\mathbf{InfBc}^a$*-identifies a class* $\mathcal{L}$ *of r.e. languages* (written: $\mathcal{L} \subseteq \mathbf{InfBc}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{InfBc}^a$-identifies each language from $\mathcal{L}$.

(c) $\mathbf{InfBc}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{InfBc}^a(\mathbf{M})]\}$.

For $a = 0$, we often write $\mathbf{TxtEx}, \mathbf{TxtBc}, \mathbf{InfEx}, \mathbf{InfBc}$ instead of

$\mathbf{TxtEx}^0, \mathbf{TxtBc}^0, \mathbf{InfEx}^0, \mathbf{InfBc}^0$, respectively.

$\mathcal{L}$ is said to be an *indexed family* of languages iff there exists an indexing $L_0, L_1, \ldots$ of languages in $\mathcal{L}$ such that the question $x \in L_i$ is uniformly decidable (i.e., there exists a recursive function $f$ such that $f(i, x) = \chi_{L_i}(x)$).

We let $\text{INIT} = \{L \mid (\exists i)[L = \{x \mid x \leq i\}]\}$.

## 3 Learning with Negative Counterexamples

In this section we define four models of learning languages from positive data and negative counterexamples. 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 learning machines get as input two texts, one for positive data, and other for negative counterexamples.

We say that $\mathbf{M}(T, T')$ converges to a grammar $i$, iff for all but finitely many $n$, $\mathbf{M}(T[n], T'[n]) = i$.

First, we define the basic 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 9** Suppose $a \in N \cup \{*\}$.

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

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

$\mathbf{M}(T, T')$ converges to a grammar $i$ such that $W_i =^a L$.

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

(c) $\mathbf{NCEx}^a = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{NCEx}^a(\mathbf{M})]\}$.

We next consider the case when the learner gets the least negative counterexample, rather than any negative counterexample. **LNC** in the definition below stands for least negative counterexample.

**Definition 10** Suppose $a \in N \cup \{*\}$.

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

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

$\mathbf{M}(T,T')$ converges to a grammar $i$ such that $W_i =^a L$.

(b) $\mathbf{M}$ $\mathbf{LNCEx}^a$*-identifies a class* $\mathcal{L}$ *of r.e. languages* (written: $\mathcal{L} \subseteq \mathbf{LNCEx}^a(\mathbf{M})$), iff $\mathbf{M}$ $\mathbf{LNCEx}^a$-identifies each language in the class.

(c) $\mathbf{LNCEx}^a = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{LNCEx}^a(\mathbf{M})]\}$.

We next consider complexity constraints on the negative counterexample. The negative counterexample is provided only if there exists one such counterexample $\leq$ the maximum positive element seen in the input so far. This addresses some complexity constraints the teacher may have. **BNC** below stands for bounded negative counterexample.

**Definition 11** Suppose $a \in N \cup \{*\}$.

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

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

$\mathbf{M}(T,T')$ converges to a grammar $i$ such that $W_i =^a L$.

(b) $\mathbf{M}$ $\mathbf{BNCEx}^a$*-identifies a class* $\mathcal{L}$ *of r.e. languages* (written: $\mathcal{L} \subseteq \mathbf{BNCEx}^a(\mathbf{M})$), iff $\mathbf{M}$ $\mathbf{BNCEx}^a$-identifies each language in the class.

(c) $\mathbf{BNCEx}^a = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{BNCEx}^a(\mathbf{M})]\}$.

The following is a generalization of Definition 11 where the negative counterexample is within some recursive factor of maximum positive element seen so far.

Let $INCFUNC = \{h \in \mathcal{R} \mid (\forall x)[h(x) \geq x] \ \wedge \ (\forall x)[h(x) \leq h(x+1)]\})$. $INCFUNC$ is class of non-decreasing functions which are greater than the identity function.

**BFNC** below stands for bounded by a function negative counterexample.

**Definition 12** Suppose $a \in N \cup \{*\}$. Suppose $h \in INCFUNC$.

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

$$T'(n) \in S_n, \text{ if } S_n \neq \emptyset \text{ and } T'(n) = \#, \text{ if } S_n = \emptyset,$$
$$\text{where } S_n = \overline{L} \cap W_{\mathbf{M}(T[n],T'[n])} \cap \{x \mid x \leq h(\max(\text{content}(T[n])))\}$$

$\mathbf{M}(T, T')$ converges to a grammar $i$ such that $W_i =^a L$.

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

(c) $\mathbf{BF}^h\mathbf{NCEx}^a = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{BF}^h\mathbf{NCEx}^a(\mathbf{M})]\}$.

(d) $\mathbf{BFNCEx}^a = \bigcup_{h \in INCFUNC} \mathbf{BF}^h\mathbf{NCEx}^a$.

Similarly one can define $\mathbf{NCBc}^a$, $\mathbf{LNCBc}^a$, $\mathbf{BNCBc}^a$ and $\mathbf{BFNCBc}^a$ criteria of inference.

We may also similarly define variants $\mathbf{NRNCEx}^a$, $\mathbf{NRLNCEx}^a$, $\mathbf{NRBNCEx}^a$, $\mathbf{NRBFNCEx}^a$ where the learner is allowed to be non-recursive. (Prefix **NR** to a criteria denotes that learner is allowed to be non-recursive).

**Proposition 13** *Suppose* $a \in N \cup \{*\}$.

*(i)* $\mathbf{TxtEx}^a \subseteq \mathbf{BNCEx}^a \subseteq \mathbf{BF}^h\mathbf{NCEx}^a \subseteq \mathbf{NCEx}^a \subseteq \mathbf{LNCEx}^a$.

*(ii)* $\mathbf{TxtBc}^a \subseteq \mathbf{BNCBc}^a \subseteq \mathbf{BF}^h\mathbf{NCBc}^a \subseteq \mathbf{NCBc}^a \subseteq \mathbf{LNCBc}^a$.

## 4 Ex-type Learning With Negative Counterexamples

We first show an example of what can be achieved by using positive data and negative counterexamples in the context of indexed families of languages. Our theorem improves a classical folklore result that every indexed family is learnable from informants. (see for example, [LZ94]). Note that there exists an indexed family which does not belong to **TxtEx** (see [Gol67]), thus **Ex**-learning without negative counterexample is weaker than **NCEx**.

**Theorem 14** *Suppose* $\mathcal{L}$ *is an indexed family. Then* $\mathcal{L} \in \mathbf{NCEx}$.

PROOF. Suppose $L_0, L_1, \ldots$ is an indexed family. On input $(\sigma, \tau)$, $\mathbf{M}$ outputs a grammar for $L_i$, for the least $i$ such that content$(\sigma) \subseteq L_i$ and $L_i \cap$content$(\tau) = \emptyset$.

Suppose $T$ is a text for $L$, and $j$ is the least number such that $L_j = L$. Then, for all $k < j$, either

(i) $L \not\subseteq L_k$, thus for large enough $n$, content$(T[n]) \not\subseteq L_k$, and thus $\mathbf{M}$ would not output $L_k$ as its conjecture, or

(ii) $L_k - L \neq \emptyset$, thus the first time $L_k$ is output there will be a negative counterexample, and thus $L_k$ would not be conjectured by $\mathbf{M}$ thereafter.

Moreover, $L_j$ always passes both the tests (content$(T[n]) \subseteq L_j$ and none of the negative counterexamples are in $L_j$). Thus, eventually $\mathbf{M}$, on text $T$ and any sequence of valid negative counterexamples, will converge to a grammar for $L_j$. ∎

We now illustrate another difference between **NCEx** learning and **TxtEx** learning.

**Theorem 15** *Suppose $\mathcal{L} \in$ **NCEx** and $L$ is a recursive language. Then $\mathcal{L} \cup \{L\} \in$ **NCEx**.*

PROOF. Suppose $\mathcal{L}$ and $L$ are as in the hypothesis. An **NCEx**-learner can learn $\mathcal{L} \cup \{L\}$ as follows. It first outputs a grammar for $L$ and waits until it receives:

(i) a negative counterexample or

(ii) an element in the text not belonging to $L$ (note that this can be recursively checked as $L$ is recursive).

If none of above happens, then clearly input language must be $L$ and the learner identifies it. If one of (i) or (ii) succeeds, then the learner continues with the strategy to **NCEx**-identify $\mathcal{L}$. It follows that $\mathcal{L} \cup \{L\} \in$ **NCEx**. ∎

Note that the above does not hold for **TxtEx**-identification as $\{F \mid F$ is finite$\} \cup \{L\} \notin$ **TxtEx** for any infinite language $L$ [Gol67].

Also note that Theorem 15 does not generalize to taking r. e. language (instead of recursive language) $L$, as witnessed by following proposition.

**Proposition 16** *Let $A$ be any recursively enumerable, but not recursive set. Let $\mathcal{L} = \{\{A \cup \{x\}\} \mid x \notin A\}$. Then, $\mathcal{L} \in$ **TxtEx**, but $\mathcal{L} \cup \{A\} \notin$ **LNCEx**.*

PROOF. It is easy to verify that $\mathcal{L} \in \mathbf{TxtEx}$, as one can search for an element $x$ in the input language which does not belong to $A$.

We now show that $\mathcal{L} \cup \{A\} \notin \mathbf{LNCEx}$. Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{LNCEx}$-identifies $\mathcal{L} \cup \{A\}$. Then, as $\mathbf{M}$ $\mathbf{LNCEx}$-identifies $A$, there exists a $\sigma, \sigma'$ such that (i) content$(\sigma) \subseteq A$, (ii) content$(\sigma') \subseteq \overline{A}$, (iii) $|\sigma| = |\sigma'|$, (iv) for $s < |\sigma|$, $\sigma'(s) = \min(W_{\mathbf{M}(\sigma[s],\sigma'[s])} - A)$, if $W_{\mathbf{M}(\sigma[s],\sigma'[s])} \not\subseteq A$, and $\#$ otherwise, (v) $\mathbf{M}(\sigma, \sigma')$ is a grammar for $A$, and (vi) $\mathbf{M}(\sigma\tau, \sigma'\#^{|\tau|}) = \mathbf{M}(\sigma, \sigma')$, for all $\tau$, with content$(\tau) \subseteq A$. (Otherwise, one can construct a text $T$ for $A$ such that $\mathbf{M}$ does not $\mathbf{LNCEx}$-identify $A$ from text $T$ (and corresponding least negative counterexamples for non-subset conjectures)).

On the other hand, for every $x \notin A \cup$content$(\sigma')$, there exists a $\tau$, content$(\tau) \subseteq A \cup \{x\}$, such that $\mathbf{M}(\sigma\tau, \sigma'\#^{|\tau|}) \neq \mathbf{M}(\sigma, \sigma')$ (otherwise, $\mathbf{M}$ does not $\mathbf{LNCEx}$-identify $A \cup \{x\}$).

Thus, for any $x \notin$ content$(\sigma')$, $x \notin A$ iff there exists a $\tau$, content$(\tau) \subseteq A \cup \{x\}$, such that $\mathbf{M}(\sigma\tau, \sigma'\#^{|\tau|}) \neq \mathbf{M}(\sigma, \sigma')$. However this contradicts the fact that $A$ is not recursive. ∎

**Proposition 17** *Suppose $a \in N \cup \{*\}$. Suppose $h \in INCFUNC$.*

*(a)* $\mathbf{LNCEx}^a \subseteq \mathbf{NCEx}^a$.

*(b)* $\mathbf{NCEx}^a \subseteq \mathbf{InfEx}^a$.

*(c)* $\mathbf{NCEx}^a \subseteq \mathbf{LNCEx}^a$.

*(d)* $\mathbf{BF}^h\mathbf{NCEx}^a \subseteq \mathbf{NCEx}^a$.

*(e)* $\mathbf{BNCEx}^a \subseteq \mathbf{BF}^h\mathbf{NCEx}^a$.

*(f)* $\mathbf{TxtEx}^a \subseteq \mathbf{BNCEx}^a$.

PROOF. (a) Note that, for any grammar $g$, one can get the least negative counterexample from an arbitrary negative counterexample $y$ by conjecturing grammars for the following languages: $W_g \cap \{x \mid x \leq z\}$, for all different values of $z \leq y$. Note that this search introduces finitely many extra mind changes; however this is ok, since in $\mathbf{Ex}$-type learning, the successful learner is allowed to make finitely many mind changes.

(b) Note that from an informant one can determine in the limit a negative counterexample, if any, for any grammar $i$. Since for $\mathbf{Ex}$-type learning the learner only makes finitely many conjectures, part (b) follows.

(c), (d), (e) and (f) easily follow from relevant definitions. ∎

The following corollary shows that using least negative counterexamples, rather than arbitrary negative counterexamples, does not enhance power of a learner - this is applicable also in case when a learner can make a finite bounded number of mistakes in the final conjecture.

**Corollary 18** *Suppose* $a \in N \cup \{*\}$*. Then,* $\mathbf{NCEx}^a = \mathbf{LNCEx}^a$*.*

The next result shows that sometimes negative counterexamples are not enough: to learn a language, the learner must have access to *all* negative examples. (In particular, it demonstrates a limitation on the learning power of our basic model).

**Theorem 19** $\mathbf{InfEx} - \mathbf{NCEx}^* \neq \emptyset$*.*

The above result follows from Theorem 45 and Theorem 46.

We now show the error hierarchy for **NCEx**-learning. That is, learning with at most $n+1$ errors in almost all conjectures in our basic model is stronger than learning with at most $n$ errors. The hierarchy easily follows from the following theorem.

**Theorem 20** *Suppose* $n \in N$*.*

*(a)* $\mathbf{TxtEx}^{n+1} - \mathbf{NCEx}^n \neq \emptyset$*.*

*(b)* $\mathbf{TxtEx}^* - \bigcup_{n\in N} \mathbf{NCEx}^n \neq \emptyset$*.*

Proof. (a) Follows from $\mathbf{TxtEx}^{n+1} - \mathbf{InfEx}^n \neq \emptyset$ [CL82] and Proposition 17(b).

(b) Follows from $\mathbf{TxtEx}^* - \bigcup_{n\in N} \mathbf{InfEx}^n \neq \emptyset$ [CL82] and Proposition 17(b). ∎

As, by Proposition 17, $\mathbf{TxtEx}^{n+1} \subseteq \mathbf{BNCEx}^{n+1} \subseteq \mathbf{BF}^h\mathbf{NCEx}^{n+1} \subseteq \mathbf{NCEx}^{n+1} \subseteq \mathbf{LNCEx}^{n+1}$, the following corollary follows from Theorem 20.

**Corollary 21** *Suppose* $n \in N$ *and* $h \in INCFUNC$*.*

*(a)* $\mathbf{NCEx}^n \subsetneq \mathbf{NCEx}^{n+1}$*.*

*(b)* $\mathbf{LNCEx}^n \subsetneq \mathbf{LNCEx}^{n+1}$*.*

*(c)* $\mathbf{BNCEx}^n \subsetneq \mathbf{BNCEx}^{n+1}$*.*

*(d)* $\mathbf{BF}^h\mathbf{NCEx}^n \subsetneq \mathbf{BF}^h\mathbf{NCEx}^{n+1}$*.*

Now we demonstrate yet another limitation on the learning power of our basic model when an arbitrary finite number of errors is allowed in the final conjecture: there are languages learnable within the classical **Bc**-style model (without negative counterexamples) and not learnable in the above variant of our basic model.

**Theorem 22** $\mathbf{TxtBc} - \mathbf{NCEx}^* \neq \emptyset$.

Proof. Follows from $\mathbf{TxtBc}-\mathbf{InfEx}^* \neq \emptyset$ [CL82] and Proposition 17(b). ∎

We will use the following proposition in some of our theorems.

**Proposition 23** *[Gol67] Suppose $L_0, L_1, \ldots$ and $L$ are such that (i) for all $i$, $L_i \subsetneq L_{i+1}$, and (ii) $\bigcup_{i \in N} L_i = L$. Then, $\mathcal{L} = \{L\} \cup \{L_i \mid i \in N\} \notin \mathbf{TxtBc}^*$ (even if one allows non-recursive learners).*

Now we turn to models where size of negative counterexamples is restricted: **BNCEx** and $\mathbf{BF}^h\mathbf{NCEx}$.

We first show that there are classes of languages learnable in our basic model that cannot be learned in any of the models that use negative counterexamples of limited size - even if the learners in the latter models are non-computable.

**Theorem 24** $\mathbf{NCEx} - \bigcup_{h \in INCFUNC} \mathbf{NRBF}^h\mathbf{NCEx}^* \neq \emptyset$.

Proof. We assume without loss of generality that pairing function is increasing in both its arguments. For $\varphi_i \in INCFUNC$, let $x_i^0 = \langle i, 0 \rangle$. Let $x_i^{j+1} = \langle i, \varphi_i(x_i^j) + 1 \rangle$. Now let $L_i^k = \{\langle i, x_i^j \rangle \mid j \leq k\}$, and $L_i^N = \{\langle i, x_i^j \rangle \mid j \in N\}$. For $\varphi_i \in INCFUNC$, let $\mathcal{L}_i = \{L_i^k \mid k \in N\} \cup \{L_i^N\}$. Let $\mathcal{L} = \bigcup_{i \in \{z \mid \varphi_z \in INCFUNC\}} \mathcal{L}_i$.

Now we show that $\mathcal{L} \in \mathbf{NCEx}$. A learner can first determine $i$ such that input is a language from class $\mathcal{L}_i$. Then the learner can output a grammar for $L_i^N$. If there is no negative counterexample to this conjecture, then we are done; otherwise the learner can follow the strategy of learning finite languages from text to learn the input language.

We now claim that $\mathcal{L}_i \notin \mathbf{NRBF}^{\varphi_i}\mathbf{NCEx}^*$, for any $\varphi_i \in INCFUNC$. To see this, note that for learning languages in $\mathcal{L}_i$, according to criterion $\mathbf{NRBF}^{\varphi_i}\mathbf{NCEx}^*$, the negative information is not useful as every $L_i^k \subseteq L_i^N$ and $\min(L_i^N - L_i^k) > \varphi_i(\max(L_i^k))$. Thus, $\mathcal{L}_i \notin \mathbf{NRBF}^{\varphi_i}\mathbf{NCEx}^*$, follows from $\mathcal{L}_i \notin \mathbf{TxtEx}^*$, even for non-recursive learner (this follows by Proposition 23, as for all $k$, $L_i^k \subsetneq L_i^{k+1}$ and $\bigcup_{k \in N} L_i^k = L_i^N$). ∎

However, the following theorem shows that if attention is restricted to only infinite languages, then **NCEx** and **BNCEx** behave similarly.

**Theorem 25** *Suppose $\mathcal{L}$ consists of only infinite languages. Then $\mathcal{L} \in \mathbf{NCEx}^a$ iff $\mathcal{L} \in \mathbf{BNCEx}^a$.*

PROOF. As $\mathbf{BNCEx}^a \subseteq \mathbf{NCEx}^a$, it suffices to show that if $\mathcal{L} \in \mathbf{NCEx}^a$ then $\mathcal{L} \in \mathbf{BNCEx}^a$. Suppose $\mathbf{M}$ $\mathbf{NCEx}^a$-identifies $\mathcal{L}$. Define $\mathbf{M}'$ as follows. $\mathbf{M}'$ on the input text $T$ of positive data for an infinite language $L$ behaves as follows. Initially let Cntrexmpls $= \emptyset$. Intuitively, Cntrexmpls denotes the set of negative counterexamples received so far. Initially let NegSet $= \emptyset$. Intuitively, NegSet denotes the set of grammars for which we know a negative counterexample. For $j \in$ NegSet, $ncex(j)$ would denote a negative counterexample for $j$. For ease of presentation, we will let $\mathbf{M}'$ output more than one conjecture (one after another) at some input point and get negative counterexamples for each of them. This is for ease of presentation and one can always spread out the conjectures.

Stage $s$ of $\mathbf{M}'$ (after seeing $T[s]$)

1. Simulate $\mathbf{M}$ on $T[s]$, by giving negative counterexamples to any conjectures $j \in$ NegSet by $ncex(j)$. Other grammars get # as counterexample.
2. Let $S$ be the set of conjectures output by $\mathbf{M}$, in the above simulation, on initial segments of $T[s]$, and let $k$ be the final conjecture.
3. If $k \notin$ NegSet, output a grammar for $\bigcup_{i \in S - \text{NegSet}} W_i$,
   Otherwise (i.e., if $k \in$ NegSet), output a grammar for $[(W_k - \text{Cntrexmpls}) \cup \bigcup_{i \in S - \text{NegSet}} W_i]$.
4. If there is no negative counterexample, then go to stage $s+1$.
5. Else (i.e., there is a negative counterexample) output one by one, for each $i \in S -$ NegSet, grammar $i$. If a negative counterexample is obtained, then place $i$ in NegSet and define $ncex(i)$ to be this negative counterexample.
   (Note that since $\mathbf{M}'$ is for **BNCEx**-type learning, negative examples received would be $\leq \max(\text{content}(T[s]))$, if any).
   Update Cntrexmpls based on new negative counterexamples obtained.
6. Go to stage $s+1$.

End Stage $s$.

Now let $T$ be a text for an infinite language $L \in \mathcal{L}$. Let NegSet$^f$ denote the set of all elements which are ever placed in NegSet in the above construction. For the text $T$, let $Neg_T$ denote the text for negative counterexamples generated as follows:

$$Neg_T(i) = \begin{cases} ncex(\mathbf{M}(T[i], Neg_T[i])), & \text{if } \mathbf{M}(T[i], Neg_T[i]) \in \text{NegSet}^f; \\ \#, & \text{otherwise.} \end{cases}$$

We claim that $Neg_T$ denotes a correct text for negative counterexamples (for $\mathbf{NCEx}^a$-model of learning) when $\mathbf{M}$ is fed $T$ as the positive data text. Clearly, if a negative counterexample is provided above then it is correct. So we only need to consider if there exists an $i$ such that $\mathbf{M}(T[i], Neg_T[i]) \notin \mathrm{NegSet}^f$, but $W_{\mathbf{M}(T[i],Neg_T[i])} \not\subseteq L$. We claim that this is not possible. To see this suppose, by way of contradiction, that $i$ is the least number for which this happens. Then, beyond some stage (by which stage all $\mathbf{M}(T[j], Neg_T[j])$ $j < i$, such that $W_{\mathbf{M}(T[j],Neg_T[j])} \not\subseteq L$, have been placed in NegSet), we have that the above construction will output a grammar which enumerates at least $W_{\mathbf{M}(T[i],Neg_T[i])}$ (see step 3). Thus, eventually a negative counterexample to $W_{\mathbf{M}(T[i],Neg_T[i])}$ would appear due to steps 3 and 5 (as the data in the input text is unbounded, due to $L$ being infinite set). A contradiction. Thus, $Neg_T$ denotes a correct sequence of negative counterexamples to $\mathbf{M}$ on text $T$.

Thus, since $\mathbf{M}$ converges on $(T, Neg_T)$, we have that for all but finitely many stages, the simulation of $\mathbf{M}$ in step 1 is correct (i.e., $\mathbf{M}'$ provides the correct negative counterexamples, if any, in the simulation). Thus, for all but finitely many stages, as $\mathbf{M}$ $\mathbf{NCEx}^a$-identifies $L$, the grammar output in step 3 by $\mathbf{M}'$ would be correct (except for possibly $a$ errors of omission, as done by the final grammar of $\mathbf{M}$) and $\mathbf{M}'$ $\mathbf{BNCEx}^a$-identifies $L$. ▮

Our next result shows that the model **BNCEx**, while being weaker than our basic model, is still quite powerful: there are classes of languages learnable in this model that cannot be learned in the classical **Bc**-style model even when an arbitrary finite number of errors is allowed in almost all conjectures.

**Theorem 26** $\mathbf{BNCEx} - \mathbf{TxtBc}^* \neq \emptyset$.

PROOF. Let $E = \{2x \mid x \in N\}$, the set of even numbers. Let $L_n = E \cup \{x \mid x \leq n\}$. Consider the class $\mathcal{L} = \{N\} \cup \{L_n \mid n \in N\}$. Clearly, $\mathcal{L} \in \mathbf{BNCEx}$ as one can output a grammar for $N$ until, if ever, there is a negative counterexample. If and when a negative counterexample is received for $N$, one can then follow the strategy to learn $\{L_n \mid n \in N\}$ (which is learnable from text alone). However, as $L_1 \subsetneq L_3 \subsetneq L_5 \ldots$ and $\bigcup_{i \in N} L_{2i+1} = N$, we have from Proposition 23 that $\mathcal{L} \notin \mathbf{TxtBc}^*$ (even by non-computable learners). ▮

The next result shows that $\mathbf{BF}^h\mathbf{NCEx}$ allows one to learn a class which is not learnable in the **BNCEx** model, even by a non-computable learner.

**Theorem 27** *Suppose $h$ is such that for all $x$, $h(x) > x$. Then* $\mathbf{BF}^h\mathbf{NCEx} - \mathbf{NRBNCEx}^* \neq \emptyset$.

PROOF. Consider $\mathcal{L} = \mathrm{INIT} \cup \{N\}$. We first show that $\mathcal{L} \in \mathbf{BF}^h\mathbf{NCEx}$. A learner can output a grammar for $N$ until, if ever, there is a negative counterexample. (Note that if the input language is $\{x \mid x \leq n\}$, for some $n$, then elements in $\{x \mid n < x \leq h(n)\} \neq \emptyset$, are valid negative counterexamples for the language $N$, once element $n$ appears in the input). If and when a negative counterexample is received for $N$, one can then follow the strategy to learn INIT, which is learnable from text alone.

On the other hand, for $\mathbf{NRBNCEx}^*$ learnability, there is never a negative counterexample, as none of the languages in the class have a negative counterexample $\leq$ maximum element present in the input. Thus, using Proposition 23, we have that $\mathcal{L} \notin \mathbf{NRBNCEx}^*$. ∎

Now we show a hierarchy for $\mathbf{BF}^h\mathbf{NCEx}$-style learning. If $h'$ is greater than $h$ in just infinitely many points, then $\mathbf{BF}^{h'}\mathbf{NCEx}$ contains languages not learnable in $\mathbf{BF}^h\mathbf{NCEx}$, even if a $\mathbf{BF}^h\mathbf{NCEx}$-learner is non-computable.

**Theorem 28** *Suppose $h, h' \in INCFUNC$. Suppose further that $x_0, x_1, \ldots$ is a recursive sequence of increasing numbers such that*

*(i) $x_0 = 0$*

*(ii) for all $i$, $h(x_{2i+1}) < x_{2i+2} \leq h'(x_{2i+1})$.*

*Let $L_N = \{x \mid (\exists i)[x_{2i} \leq x \leq x_{2i+1}]\}$.*

*Let $L_j = \{x \mid (\exists i \leq j)[x_{2i} \leq x \leq x_{2i+1}]\}$.*

*Then, $\mathcal{L} = \{L_j \mid j \in N\} \cup \{L_N\} \in \mathbf{BF}^{h'}\mathbf{NCEx} - \mathbf{NRBF}^h\mathbf{NCEx}^*$.*

Proof. Clearly $\mathcal{L} \in \mathbf{BF}^{h'}\mathbf{NCEx}$, as one can output a grammar for $L_N$ until, if ever, a negative counterexample is received. (Note that if input language is $L_j$, then eventually there exists such a negative counterexample as $x_{2j+2} \in L_N - L_j$ and $x_{2j+2} \leq h'(x_{2j+1})$.) If and when a negative counterexample is received, the learner can then follow the learning strategy (similar to that for INIT) for $\{L_j \mid j \in N\}$ (which can be learned from text alone).

On the other hand, for $\mathbf{NRBF}^h\mathbf{NCEx}^*$ learnability there is never a negative counterexample from the set $L_N$ due to the fact that $\min(L_N - L_j) > h(\max(L_j))$ for any $j \in N$. This essentially renders the negative information useless. Thus $\mathcal{L} \in \mathbf{NRBF}^h\mathbf{NCEx}^*$ would mean $\mathcal{L} \in \mathbf{TxtEx}^*$ (by non-computable learner), which is not true by Proposition 23 (as $L_j \subsetneq L_{j+1}$ and $\bigcup_{j \in N} L_j = L_N$). ∎

**Corollary 29** *Suppose $h, h' \in INCFUNC$ such that $h'(x) > h(x)$ for infinitely many $x$. Then $\mathbf{BF}^{h'}\mathbf{NCEx} - \mathbf{NRBF}^h\mathbf{NCEx}^* \neq \emptyset$.*

Proof. Note that, for any pair of recursive functions $h$ and $h'$ such that $h'(x) > h(x)$ for infinitely many $x$, one can define a recursive sequence of $x_i$ such that hypotheses (i) and (ii) in Theorem 28 hold. This can be done by taking $x_0 = 0$ and inductively defining $x_{2i+1}, x_{2i+2}$ such that $x_{2i+1} > x_{2i}$ and $h(x_{2i+1}) < h'(x_{2i+1}) = x_{2i+2}$. Now corollary follows from Theorem 28. ∎

On the other hand, if $h'(x) \leq h(x)$ for all but finitely many $x$, then clearly $\mathbf{BF}^{h'}\mathbf{NCEx}^a \subseteq \mathbf{BF}^h\mathbf{NCEx}^a$.

We now turn our attention to the power of non-computable learners. The following proposition follows from definitions.

**Proposition 30** *Suppose $a \in N$.*

*(a)* $\mathbf{NRNCEx}^a \subseteq \mathbf{NRLNCEx}^a$.

*(b)* $\mathbf{NRBFNCEx}^a \subseteq \mathbf{NRNCEx}^a$.

*(c)* $\mathbf{NRBNCEx}^a \subseteq \mathbf{NRBFNCEx}^a$.

As the next result shows, a non-computable learner in our basic model has the "ultimate power": it can learn all recursively enumerable languages.

**Theorem 31** $\mathcal{E} \in \mathbf{NRNCEx}$.

PROOF. A non-effective learner can search for the least grammar $i$ such that content$(T) \subseteq W_i$ and $i$ does not generate a negative counterexample. Thus, $\mathcal{E} \in \mathbf{NRNCEx}$. ∎

As $\mathcal{E} \in \mathbf{NRInfEx}$, we have

**Corollary 32** $\mathbf{NRNCEx} = \mathbf{NRInfEx}$.

**Theorem 33** *Suppose $\mathcal{L}$ is such that:*

*for any infinite $L \in \mathcal{L}$, there exist only finitely many $n$ such that $L \cap \{x \mid x < n\} \in \mathcal{L}$.*

*Then, $\mathcal{L} \in \mathbf{NRBNCEx}$.*

PROOF. Define (possibly nonrecursive) $\mathbf{M}$ as follows. On input $(T[n], T'[n])$, output the least $i$ such that $W_i \in \mathcal{L}$, and

(i) content$(T[n]) \subseteq W_i$, and

(ii) content$(T'[n]) \cap W_i = \emptyset$, and

(iii) $W_i =$ content$(T[n])$ or content$(T[n]) \notin \mathcal{L}$ or for all $t$, [content$(T[n]) \neq W_i \cap \{x \mid x < t\}$].

We claim that above $\mathbf{M}$ would **NRBNCEx**-identify $\mathcal{L}$. To see this suppose $T$ is a text for $L \in \mathcal{L}$. Let $i$ be the least grammar for $L$. If $L$ is finite, then let $n$ be such that content$(T[n]) = L$. If $L$ is infinite, then let $t$ be such that for all $t' \geq t$, $W_i \cap \{x \mid x < t'\} \notin \mathcal{L}$; then let $n$ be such that $W_i \cap \{x \mid x < t\} \subseteq$ content$(T[n])$. Now, for $n' \geq n$, $i$ satisfies conditions (i)—(iii) above. Thus, $\mathbf{M}(T[n']) \leq i$.

Now, for any $j < i$, let $n' > n$ be so large that:

(iv) if $L \not\subseteq W_j$, then $T[n'] \not\subseteq W_j$,

and

(v) if $\min(W_j - L) \leq \max(L)$, then $\min(W_j - L) \leq \max(T[n'])$.

Note that there exists such an $n'$.

We claim that, any $j < i$ appears at most once as a conjecture beyond $T[n']$. Clearly, if $L \not\subseteq W_j$, then $j$ cannot appear as $\mathbf{M}$'s conjecture beyond $T[n']$ due to (i) and (iv) above. Furthermore, if $\min(W_j - L) \leq \max(L)$, then $j$ appears at most once beyond $T[n']$ (as then we will get a negative counterexample for conjecture $j$). Thus, for $j$ to appear more than once beyond $T[n']$, we must have $L \subseteq W_j$ and $\min(W_j - L) > \max(L)$. But then $L$ is finite, and $W_j \cap \{x \mid x < \max(L)\} = L$. Thus, (iii) above implies that $j$ would not be output by $\mathbf{M}$ beyond $T[n']$.

Also, since $i$ satisfies (i)—(iii) above, for almost all $n'' \geq n'$, we would have that $\mathbf{M}$ outputs $i$ on $T[n'']$. ∎

**Theorem 34** *Suppose $\mathcal{L}$ is such that:*

*there exists an infinite $L \in \mathcal{L}$, there exist infinitely many $n$ such that $L \cap \{x \mid x < n\} \in \mathcal{L}$.*

*Then, $\mathcal{L} \notin$ **NRBNCBc**$^*$.*

Proof. Suppose by way of contradiction that $\mathbf{M}$ **NRBNCBc**$^*$-identifies $\mathcal{L}$, when least eligible negative counterexample ($\leq$ maximum positive element in the input data), if any, is given to $\mathbf{M}$. Suppose $\sigma$ is such that (i) content$(\sigma) \subseteq L$, and (ii) for all $\tau$ such that content$(\tau) \subseteq L$, $\mathbf{M}$ outputs a grammar for a finite variant of $L$ on input positive data $\sigma\tau$ (where the negative counterexamples provided are the least eligible counterexample as described above). Note that there exists such a $\sigma$, since otherwise we can inductively build a text $T$ for $L$ on which $\mathbf{M}$ fails to **NRBNCBc**$^*$-identify $L$.

Let $t$ be such that content$(\sigma) \subseteq L \cap \{x \mid x < t\}$ and $L \cap \{x \mid x < t\} \in \mathcal{L}$. Now, $\mathbf{M}$ does not **NRBNCBc**$^*$-identify $L \cap \{x \mid x < t\}$, on any text $T$, extending

$\sigma$, for $L \cap \{x \mid x < t\}$ (since all its conjectures beyond $\sigma$ on $T$ are for finite variants of $L$). ∎

Note that Theorems 33 and 34 give a characterization for **NRNCEx**-identification and also show that **NRBNCEx** = **NRBNCBc**$^*$.

One can similarly show:

**Theorem 35** *Fix $h \in INCFUNC$. Suppose $\mathcal{L}$ is such that:*

*for any infinite $L \in \mathcal{L}$, there exist only finitely many $n \in L$ such that $L \cap \{x \mid x \leq n\} \in \mathcal{L}$, but $\min(L - \{x \mid x \leq n\}) > h(n)$.*

*Then, $\mathcal{L} \in \mathbf{NRBF}^h\mathbf{NCEx}$.*

**Theorem 36** *Fix $h \in INCFUNC$. Suppose $\mathcal{L}$ is such that:*

*there exists an infinite $L \in \mathcal{L}$, for infinitely many $n \in L$, $L \cap \{x \mid x \leq n\} \in \mathcal{L}$, and $\min(L - \{x \mid x \leq n\}) > h(n)$.*

*Then, $\mathcal{L} \notin \mathbf{NRBF}^h\mathbf{NCEx}$.*

## 5 Locking Sequence and Delayed Counterexamples

In this section we introduce the concept of a *locking sequence* for our **Ex**-style learning model. Locking sequence (see [BB75]) is an important tool in understanding and characterizing learning languages in the limit. Informally, a locking sequence is an initial fragment of the input text that is sufficient for a learner to identify the target language. Once the locking sequence has been inputted, the learner never changes its mind. Using the concept of locking sequence, we obtain a characterization of **NCEx**-type learning. Our concept of locking sequence turns out to be very useful in our following discussion of learning from positive data and negative counterexamples when counterexamples can be delayed.

For the following, we will often consider giving machine **M** least valid negative information, if any. To this end define $neginput_{\mathbf{M},L,\sigma}$ as follows. For $n < |\sigma|$,

$$neginput_{\mathbf{M},L,\sigma}(n) = \begin{cases} \#, & \text{if } W_{\mathbf{M}(\sigma[n], neginput_{\mathbf{M},L,\sigma}[n])} \subseteq L; \\ x, & \text{otherwise,} \\ & \text{where } x = \min(W_{\mathbf{M}(\sigma[n], neginput_{\mathbf{M},L,\sigma}[n])} - L). \end{cases}$$

(for **BNCEx**, $\mathbf{BF}^h\mathbf{NCEx}$-identification the first clause above is appropriately modified to check containment only for elements $\leq \max(\text{content}(\sigma))$ or $h(\max(\text{content}(\sigma)))$ respectively).

When the input language $L$ is implicit, we also define $\mathbf{LN_M}(\sigma) = \mathbf{M}(\sigma, neginput_{\mathbf{M},L,\sigma}[|\sigma|])$.

Intuitively, **LN** above stands for least negative relevant counterexample given.

**Definition 37** $(\sigma, j)$-is said to be a **NCEx**-*stabilizing sequence for* **M** *on* $L$ iff

(i) $\text{content}(\sigma) \subseteq L$,

(ii) $W_{\mathbf{LN_M}(\sigma)} \subseteq L$,

(iii) For all $\tau$ such that $\text{content}(\tau) \subseteq L$, $\mathbf{LN_M}(\sigma) = \mathbf{LN_M}(\sigma\tau)$.

(iv) For all $n \leq |\sigma|$, $\min(W_{\mathbf{LN_M}(\sigma[n])} \cap \overline{L}) = \min(W_{\mathbf{LN_M}(\sigma[n]),j} \cap \overline{L})$.

(For **BNCEx**-stabilizing sequence above definition is appropriately modified by changing (ii) and (iv) to

(ii') $W_{\mathbf{LN_M}(\sigma)} \cap \{x \mid x \leq \max(L)\} \subseteq L$,

(iv') For all $n \leq |\sigma|$, $\min(W_{\mathbf{LN_M}(\sigma[n])} \cap \overline{L} \cap \{x \mid x \leq \max(\text{content}(\sigma[n]))\}) = \min(W_{\mathbf{LN_M}(\sigma[n]),j} \cap \overline{L} \cap \{x \mid x \leq \max(\text{content}(\sigma[n]))\})$. )

**Remark**: Recall that $W_{i,j} = \{x \mid x < j \ \wedge \ \Phi_i(x) < j\}$. Thus $W_{i,j} \subseteq \{x \mid x < j\}$.

Note that if $(\tau, j)$ is a stabilizing sequence for **M** on $L$, then so is $(\tau', j')$, for any $j' \geq j$, and $\tau' \supseteq \tau$ with $\text{content}(\tau') \subseteq L$.

**Definition 38** $(\sigma, j)$-is said to be a **NCEx**-*locking sequence for* **M** *on* $L$ iff

(i) $(\sigma, j)$ is a stabilizing sequence for **M** on $L$, and

(ii) $\mathbf{LN_M}(\sigma)$ is a grammar for $L$.

**Proposition 39** *Suppose* **M** **NCEx**-*identifies* $L$. *Then*

*(a) there exists a* **NCEx**-*stabilizing sequence for* **M** *on* $L$, *and*

*(b) every* **NCEx**-*stabilizing sequence for* **M** *on* $L$ *is also a* **NCEx**-*locking sequence for* **M** *on* $L$.

Proof. (a) Consider the process in which **M** is always given the least negative counterexample, if any. We first claim that there exists a $\tau$ such that (i) content($\tau$) $\subseteq L$, (ii) $W_{\mathbf{LN_M}(\tau)} \subseteq L$, and (iii) for any $\tau' \supseteq \tau$ such that content($\tau'$) $\subseteq L$, $\mathbf{LN_M}(\tau) = \mathbf{LN_M}(\tau')$. This follows immediately from the fact that **M** **NCEx**-identifies $L$. (Otherwise, one can construct a text $T$ such that (I) **M** does not converge on $T$ or (II) **M** makes infinitely many wrong conjectures on $T$. To see this, let $T'$ be a text for $L$. Define $\tau_i$ as follows. $\tau_0$ is a sequence consisting of just $T'(0)$. Inductively define $\tau_{i+1}$ as follows. If $\tau_i$ does not satisfy the requirements (i)–(iii), then either (I') for some $\sigma$ extending $\tau_i$ with content($\sigma$) $\subseteq L$, $\mathbf{LN_M}(\tau_i) \neq \mathbf{LN_M}(\sigma)$, or (II') $W_{\mathbf{LN_M}(\tau_i)}$ contains an element outside $L$ — in this case let $\sigma = \tau_i$. Now let $\tau_{i+1} = \sigma T'(i+1)$. It immediately follows that $\bigcup_{i \in N} \tau_i$ is a text for $L$, and $\mathbf{LN_M}(T)$ makes infinitely many mind changes on $T$ or makes infinitely many wrong conjectures on $T$.)

Now define $j$ to be the least value such that, for all $n \leq |\tau|$, $\min(W_{\mathbf{LN_M}(\tau[n])} \cap \overline{L}) = \min(W_{\mathbf{LN_M}(\tau[n]),j} \cap \overline{L})$.

Now, $(\tau, j)$ satisfies the definition of being **NCEx**-stabilizing sequence for **M** on $L$.

(b) Follows from definition of **NCEx**-identification. ∎

The following proposition demonstrates how learning in our basic model can be characterized using locking sequences.

**Proposition 40** $\mathcal{L} \in$ **NCEx** *iff there exists an* **M** *such that for each* $L \in \mathcal{L}$,

*(a) there exists a* **NCEx**-*stabilizing sequence for* **M** *on* $L$, *and*

*(b) every stabilizing sequence for* **M** *on* $L$ *is a* **NCEx**-*locking sequence for* **M** *on* $L$.

PROOF. Left to Right direction follows from Proposition 39.

For right to left part note that a **NCEx**-learner $\mathbf{M}'$ can search for a $(\sigma, j)$ such that the following four properties are satisfied:

(i) content($\sigma$) $\subseteq L$;

(ii) $W_{\mathbf{LN_M}(\sigma)} \subseteq L$;

(iii) For all $\tau$ such that content($\tau$) $\subseteq L$, $\mathbf{LN_M}(\sigma) = \mathbf{LN_M}(\sigma\tau)$;

(iv) For all $n \leq |\sigma|$, $\min(W_{\mathbf{LN_M}(\sigma[n])} \cap \overline{L}) = \min(W_{\mathbf{LN_M}(\sigma[n]),j} \cap \overline{L})$.

Note that a **NCEx** learner can determine $\chi_L[j]$ (by conjecturing grammars for $\{i\}$, $i \leq j$). Thus, second part of the equality in (iv) above can be effectively determined, and thus any violation of (iv) can be determined in r.e. sense (by outputting $\mathbf{LN_M}(\sigma[n])$ and, if there is a negative counterexample, enumerating $W_{\mathbf{LN_M}(\sigma[n])}$ and checking the elements in $\overline{L} \cap \{x \mid x < j\}$). Assuming (iv) holds, negative information needed for calculating $W_{\mathbf{LN_M}}(\tau)$, for $\tau \subseteq \sigma$, can also be effectively found using $\chi_L[j]$. Violation of (ii) can be determined by checking if conjecturing $\mathbf{LN_M}(\sigma)$ leads to a negative example. Violation of (iii) is easy to check. Thus, in the limit, we can find a stabilizing sequence for $\mathbf{M}$ on $L$, if any. Hence, a learner can output $\mathbf{LN_M}(\sigma)$, in the limit, for one such stabilizing sequence (which gives a grammar for $L$ by clause (b)). ∎

For **BNCEx**-identification we have the following characterization. The proof is similar to the above proposition, except that we need a slight modification as one may not be able to determine $\chi_L[j]$ from the input (due to extra constraints on negative examples). Similar characterization results can be proved for $\mathbf{BF}^h\mathbf{NCEx}$-identification also.

**Proposition 41** $\mathcal{L} \in \mathbf{BNCEx}$ *iff there exists an* $\mathbf{M}$ *such that for each* $L \in \mathcal{L}$*,*

*(a) there exists a* **BNCEx***-stabilizing sequence for* $\mathbf{M}$ *on* $L$*, and*

*(b) every stabilizing sequence for* $\mathbf{M}$ *on* $L$ *is a* **BNCEx***-locking sequence for* $\mathbf{M}$ *on* $L$*.*

Proof. Left to Right direction follows by using an analogue of Proposition 39.

For Right to Left direction we proceed similar to Proposition 40, except that we need to be careful in the sense that one may not be able to obtain $\chi_L[j]$ from the input, if the input language is finite (due to constraints on the negative information provided).

Thus, on input $T$, a **BNCEx** learner $\mathbf{M}'$ searches for a $(\sigma, j)$ such that:

(I) content$(T) \subseteq \{x \mid x \leq j\}$, or

(II) The following four conditions hold:

(II.i) content$(\sigma) \subseteq L$;

(II.ii) $W_{\mathbf{LN_M}(\sigma)} \cap \{x \mid x \leq \max(L)\} \subseteq L$;

(II.iii) For all $\tau$ such that content$(\tau) \subseteq L$, $\mathbf{LN_M}(\sigma) = \mathbf{LN_M}(\sigma\tau)$;

(II.iv) For all $n \leq |\sigma|$, $\min(W_{\mathbf{LN_M}(\sigma[n])} \cap \overline{L} \cap \{x \mid x \leq \max(\text{content}(\sigma[n]))\}) = \min(W_{\mathbf{LN_M}(\sigma[n]),j} \cap \overline{L} \cap \{x \mid x \leq \max(\text{content}(\sigma[n]))\})$.

Note that one can do the above search by dovetailing over all pairs $(\sigma, j)$, such that each pair gets infinitely many chances, and first we check if (I) above holds, and if not, check if (II) holds. One can find in the limit a $(\sigma, j)$ such that (I) or (II) holds, if such a $(\sigma, j)$ exists. (Note that, if (I) does not hold for a particular $(\sigma, j)$, then one can determine $\chi_L[j]$, as $L$ contains an element $> j$. Thus, one can determine violation of (II) in a way similar to that done in the proof of Proposition 40).

For any candidate $(\sigma, j)$, if (I) seems to hold, $\mathbf{M}'$ outputs a grammar for $\mathrm{content}(T) \cap \{x \mid x \leq j\}$, and if case (II) seems to hold, $\mathbf{M}'$ outputs $\mathbf{LN_M}(\sigma)$. If none of (I) and (II) hold, we move on to the next candidate.

Now, due to checking of the conditions (I) and (II), $\mathbf{M}'$ can only converge to a conjecture of $\mathbf{M}$ on a stabilizing sequence (for $\mathbf{M}$ on $L$), due to success of (II) or to a conjecture for finite language due to success of (I). For $L \in \mathcal{L}$, in the former case, by condition (b) in the proposition, we have **BNCEx**-identification, and for the latter case we clearly have **BNCEx**-identification due to explicitly outputting grammar for the input language, which is finite.

Furthermore, every finite language $L$ would eventually lead to convergence due to success of (I) or earlier due to some stabilizing sequence for $\mathbf{M}$ on $L$. Also, every infinite language $L \in \mathcal{L}$, would eventually lead to convergence due to presence of some stabilizing sequence for $\mathbf{M}$ on $L$.

From above, **BNCEx** identification of $\mathcal{L}$ by $\mathbf{M}'$ follows. ▮

We now consider several variants of **NCEx** model where the negative examples may not appear immediately, nor may they appear for all conjectures enumerating a non-subset of $L$. These variants reflect complexity constraints on the teacher – yet differently from the models with limited size of negative counterexamples. As formal definitions of the models to be presented are technically rather complex, we proceed below somewhat informally.

**Definition 42** Consider the following models for delayed negative counterexamples.

**D**1: The learner eventually receives a negative counterexample for every hypothesis which enumerates a non-subset of $L$. We do not constrain when this negative counterexample appears, nor is the negative counterexample tagged with the hypothesis to which it is a counterexample.

**D**2: If the learner converges to a hypothesis, and this hypothesis enumerates a non-subset of $L$, then the learner will eventually receive a negative counterexample for it (idea here is that abandoned hypothesis may not get negative counterexample).

**D**3: Let $T$ be a text for language $L$, and let $j_m$ denote the conjecture of the learner on input $T[m]$. For all $m$, if $W_{j_m} \not\subseteq L$, then there exists a negative counterexample $x$ presented to the learner such that for some $m' \geq m$, $x \in W_{j_{m'}} - L$ (here the idea is that once a grammar gets a negative counterexample, one may consider all previously output non-subset grammars addressed).

**D**4: If a grammar is output infinitely often by the learner, and this grammar enumerates a nonsubset of $L$, then the learner eventually receives a negative counterexample for the grammar. We do not constrain when this negative counterexample appears, nor is the counterexample tagged with the hypothesis to which it is a counterexample.

Clearly, **D**2 is contained in each of **D**1, **D**3 and **D**4. Thus showing that **NCEx** = **D**2 means the collapsing of all these variants to **NCEx**. This in some sense would show that the model we have chosen is reasonably robust. The following, quite surprising result demonstrates that all the above models do collapse to **NCEx**: delays do not constrain learners if positive data and negative counterexamples are eventually available!

**Theorem 43** **NCEx** = **D**2.

Proof. Clearly, **D**2 $\subseteq$ **NCEx**. We show that **NCEx** $\subseteq$ **D**2. Suppose **M** **NCEx**-identifies $\mathcal{L}$. We define a **D**2 learner **M**$'$ as follows:

**M**$'$ tries to search for a stabilizing sequence for **M** on $L$.

**M**$'$ on the input text $T$ does the following:

**M**$'$ on text $T$ for a language $L$.
For each pair $(\sigma, j)$:
1. First determine $\chi_L[j]$. (Note that **M**$'$ can determine if a particular element $x$ is in $L$ or not, by repeatedly outputting a grammar for $\{x\}$ until it either receives $x$ in text or $x$ as negative example (one of these must happen, otherwise **M**$'$ converges to grammar for $\{x\}$ on $T$, but does not receive the required positive/negative example). Thus it can determine $\chi_L[j]$.)
2. If content$(\sigma) \not\subseteq$ content$(T)$, then go to next iteration of the loop (note that, whether content$(\sigma) \subseteq$ content$(T)$, can be determined similarly to step 1 above).
3. Else, assume the following property:
   (P1) for all $\tau \subseteq \sigma$, $\min(W_{\mathbf{LN_M}(\tau)} \cap \overline{L}) = \min(W_{\mathbf{LN_M}(\tau),j} \cap \overline{L})$ (This is property (iv) in definition of stabilizing sequence).
   and calculate $\mathbf{LN_M}(\tau)$, for $\tau \subseteq \sigma$, and
   $S = \{\mathbf{LN_M}(\tau) \mid W_{\mathbf{LN_M}(\tau)} \subseteq L\}$ (i.e., the grammars output by $\mathbf{LN_M}$ on prefixes of $\sigma$, for which # was given as negative example).

4. Output a grammar for $\bigcup_{i\in S} W_i$.
5. Idle until at least one of the following is satisfied:
   (a) there exists a negative counterexample for $\bigcup_{i\in S} W_i$. (This verifies property (ii) in the definition of stabilizing sequence and part of property (P1) above (the part where $W_{\mathbf{LN_M}(\tau)} \cap \overline{L} = \emptyset$)).
   (b) assuming (a) does not hold, check if property (P1) above is violated. (Note that this can be verified, assuming (a) above does not hold, by enumerating the elements output by $\mathbf{LN_M}(\sigma[n])$).
   (c) assuming, (a) and (b) do not hold, check if there exists a $\tau$ such that $\mathrm{content}(\tau) \subseteq L$, and $\mathbf{LN_M}(\sigma) \neq \mathbf{LN_M}(\sigma\tau)$. (This verifies property (iii) in the definition of stabilizing sequence).
6. If one of (a) to (c) succeed, then go to the next iteration of the For loop.

EndFor

Now note that, if in any iteration of the For loop, (a)–(c) do not succeed, then we have that $(\sigma, j)$ is a stabilizing sequence for $\mathbf{M}$ on $L$, and $L \supseteq \bigcup_{i\in S} W_i \supseteq W_{\mathbf{LN_M}(\sigma)} = L$. Thus, $\mathbf{M}'$ converges to correct grammar. On the other hand, if one of (a) to (c) succeed, then $(\sigma, j)$ is not a stabilizing sequence for $\mathbf{M}$ on $L$.

It is now easy to verify that, if there exists a stabilizing sequence for $\mathbf{M}$ on $L$, then for some such stabilizing sequence $(\sigma, j)$ the conditions (a)–(c) are not satisfied, and $\mathbf{M}'$ above converges to a grammar for $\bigcup_{i\in S} W_i$ (where $S$ is as in the iteration for $(\sigma, j)$).

Thus, $\mathbf{M}'$ $\mathbf{D}2$-identifies any language $\mathbf{NCEx}$-identified by $\mathbf{M}$. ∎

Note that the proof for the above theorem did not use the exact negative counterexample, but just the fact that a negative counterexample existed for the latest conjecture. In other words, our basic (and the most powerful) learning model is equivalent to the one where a learner gets only answers "yes" or "no" to the subset queries (when it is tested if the current conjecture generates a subset of the target language)!

As $\mathbf{D}4 \supseteq \mathbf{D}2$ and $\mathbf{D}3 \supseteq \mathbf{D}2$ and $\mathbf{D}1 \supseteq \mathbf{D}2$, we have that all of these are same as $\mathbf{NCEx}$.

## 6 Bc-type Learning With Negative Counterexamples

In this section we explore **Bc**-style learning from positive data and negative counterexamples. First we show that, for **Bc**-style learning, similarly to **Ex**-style learning, our basic model is equivalent to learning with the least negative counterexamples.

**Proposition 44** *(a)* **NCBc** = **LNCBc**.

*(b)* **LNCBc** $\subseteq$ **InfBc**.

PROOF. (a) Clearly, **NCBc** $\subseteq$ **LNCBc**. For **LNCBc** $\subseteq$ **NCBc**, note that, for any grammar $g$, one can get the least negative counterexample from arbitrary negative counterexample $y$ by conjecturing grammars for the following languages: $W_g \cap \{x \mid x \leq z\}$, for all different values of $z \leq y$. Note that this search introduces finitely many extra wrong conjectures, for each wrong conjecture of the **NCBc**-learner. However, since in **Bc**-type learning all but finitely many grammars output are for the input language, this does not hurt the simulation.

(b) Suppose **M** **LNCBc**-identifies $\mathcal{L}$. Define machine **M**$'$ as follows.

For an informant $I$ for $L$, define text $T$ for $L$ as follows:

$$T(i) = \begin{cases} x, & \text{if } I(x) = 1; \\ \#, & \text{otherwise.} \end{cases}$$

On input $I[n]$, output $\mathbf{M}(T[n], \tau)$, where $\tau$ is of length $n$, where for $i < n$,

$$\tau(i) = \begin{cases} \#, & \text{if } W_{\mathbf{M}(T[i],\tau[i]),n} \cap \\ & \{x < n \mid I(x) = 0\} = \emptyset; \\ \min(W_{\mathbf{M}(T[i],\tau[i]),n} \cap \{x < n \mid I(x) = 0\}), & \text{otherwise.} \end{cases}$$

Now if **M** **LNCBc**-identifies $L$, then for all but finitely many $n$, the negative answers given for conjectures of **M** on $T$ are correct and hence **M**$'$ reproduces the output of **M** on all except for finitely many initial segments of $T$. Thus, **M**$'$ **InfBc**-identifies $L$. ∎

Our next result shows that, to learn a language, sometimes even for **Bc**-style learning, positive data and negative counterexamples are not enough - the learner must have access to all negative data. In particular, limitations on the learning power of our basic **Bc**-style model are established.

**Theorem 45** **InfEx** $-$ **NCBc** $\neq \emptyset$.

PROOF. Let $\mathcal{L} = \{L \mid (\exists e)[\min(L) = 2e] \wedge$

(i) $[L = W_e$ and $(\forall x \geq e)[L \cap \{2x, 2x+1\} \neq \emptyset]]$.

OR

(ii) $(\exists x > e)[L \cap \{2x, 2x+1\} = \emptyset, \wedge (\forall y > 2x+1)[y \in L]]$

$\}$

It is easy to verify that $\mathcal{L} \in$ **InfEx**. A learner can easily find $e$ as above, and whether there exists $x > e$ such that both $2x, 2x+1$ are not in the input language. This information is sufficient to identify the input language.

We now show that $\mathcal{L} \notin$ **NCBc**. Intuitively, the idea is that for a learner which learns languages satisfying clause (ii) above, for every $\sigma$ (satisfying content$(\sigma) \subseteq \{x \mid x \geq 2e\}$), and any finite set $S$ of negative counterexamples provided to the learner (where $S$ does not contain both $2x$ and $2x+1$, for any $x$), there must exist an extension $\tau$ of $\sigma$ (satisfying content$(\tau) \subseteq \{x \mid x \geq 2e\} - S$) such that the learner, on input $\tau$, outputs a grammar for a proper extension of content$(\tau)$. This allows us to choose an appropriate element of the conjecture as negative counterexample (along with rendering the conjecture false). Iteratively, above method allows us to define $\sigma_0 \subset \sigma_1 \subset \sigma_2 \ldots$ which form larger and larger initial segments of the language $W_e$ (satisfying clause (i)) diagonalizing against **NCBc**. Note that we needed a pair $\{2x, 2x+1\}$, to separate (i) from (ii) in the definition of $\mathcal{L}$, as one of the elements may be needed for giving the negative counterexamples as mentioned above. The diagonalization below is slightly more complicated, as one may not always be able to effectively find whether a conjecture on $\sigma_s$ properly extends content$(\sigma_s)$.

We now proceed formally.

Suppose by way of contradiction that machine **M** **NCBc**-identifies $\mathcal{L}$. Then by the Kleene Recursion Theorem [Rog67] there exists a recursive function $e$ such that $W_e$ may be defined as follows.

Initially, let $W_e = \{2e, 2e+1\}$ and $\sigma_0$ be such that content$(\sigma_0) = \{2e, 2e+1\}$. Intuitively Cntrexmpls denotes the set of elements frozen to be outside the diagonalizing language being constructed. Initially, Cntrexmpls $= \{x \mid x < 2e\}$. Intuitively, NegSet is the set of conjectured grammars for which we have found a negative counterexample (in Cntrexmpls). Initially let NegSet $= \emptyset$. $ncex(j)$ is a function which gives, for $j \in$ NegSet, a negative counterexample from Cntrexmpls. For the following, let $\gamma_\tau$ be a sequence of length $|\tau|$ defined as follows. For $i < |\tau|$,

$$\gamma_\tau(i) = \begin{cases} ncex(\mathbf{M}(\tau[i], \gamma_\tau[i])), & \text{if } \mathbf{M}(\tau[i], \gamma_\tau[i]) \in \text{NegSet}; \\ \#, & \text{otherwise.} \end{cases}$$

(where the value of NegSet is as at the time of above usage).

Let $x_0 = 2e + 2$. Intuitively, $x_s$ is the least even element greater than $\max(\text{content}(\sigma_s) \cup \text{Cntrexmpls})$. Also we will have the invariant that at start of stage $s$,

(i) every element $< x_s$ is either in content($\sigma_s$) or Cntrexmpls and

(ii) content($\sigma_s$) consists of elements enumerated in $W_e$ before stage $s$.

Go to stage 0.

Stage $s$

1. Dovetail steps 2 and 3 until step 2 or 3 succeed. 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 if step 3 can succeed by simulating $\mathbf{M}(\tau, \gamma_\tau)$ for $s$ steps, then step 3 succeeded first (and for the shortest such $\tau$), otherwise whichever of these steps succeeds first is taken. (So some priority is given to step 3 in the dovetailing).
2. Search for a $\tau \supseteq \sigma_s$ such that
   content($\tau$) $\subseteq$ content($\sigma_s$) $\cup \{x \mid x \geq x_s + 2\}$,
   $\mathbf{M}(\tau, \gamma_\tau) \notin$ NegSet and
   $W_{\mathbf{M}(\tau,\gamma_\tau)}$ enumerates an element not in content($\tau$).
3. Search for a $\tau \subseteq \sigma_s$ such that $\mathbf{M}(\tau, \gamma_\tau) \notin$ NegSet and $W_{\mathbf{M}(\tau,\gamma_\tau)}$ enumerates an element not in content($\sigma_s$).
4. Let $\tau$ be as found in step 2, and $j = \mathbf{M}(\tau, \gamma_\tau)$, and $z$ be the element found to be enumerated by $W_j$ which is not in content($\tau$).
   Let NegSet = NegSet $\cup \{j\}$.
   Let Cntrexmpls = Cntrexmpls $\cup \{z\}$.
   Let $ncex(j) = z$.
   Let $x_{s+1}$ be the least even number $> \max(\text{content}(\tau) \cup \{x_s, z\})$.
   Enumerate $\{x \mid x_s \leq x < x_{s+1}\} - \{z\}$ in $W_e$.
   Let $\sigma_{s+1}$ be an extension of $\tau$ such that content($\sigma_{s+1}$) = $W_e$ enumerated until now.
   Go to stage $s + 1$.
5. Let $\tau$ be as found in step 3, and $j = \mathbf{M}(\tau, \gamma_\tau)$, and $z$ be the element found to be enumerated by $W_j$ which is not in content($\sigma_s$).
   Let NegSet = NegSet $\cup \{j\}$.
   Let Cntrexmpls = Cntrexmpls $\cup \{z\}$.
   Let $ncex(j) = z$.

Let $x_{s+1}$ be the least even number $> \max(\{x_s, z\})$.
Enumerate $\{x \mid x_s \leq x < x_{s+1}\} - \{z\}$ in $W_e$.
Let $\sigma_{s+1}$ be an extension of $\sigma_s$ such that content$(\sigma_{s+1}) = W_e$ enumerated until now.
Go to stage $s+1$.
End stage $s$

We now consider the following cases:

*Case 1*: Stage $s$ starts but does not finish.

In this case let $L = W_e \cup \{x \mid x \geq x_s + 2\}$. Note that, due to non-success of steps 2 and 3, the negative information given in computation of $\gamma_\tau$ based on NegSet is correct. Thus, for any text $T$ for $L$ extending $\sigma_s$, for $n > |\sigma_s|$, $\mathbf{M}(T[n], \gamma_{T[n]}) \in$ NegSet or it enumerates only a finite set (otherwise step 2 would succeed). Thus, $\mathbf{M}$ does not **NCBc**-identify $L$.

*Case 2*: All stages finish.

Let $L = W_e$. Let $T = \bigcup_{s \in N} \sigma_s$. Note that $T$ is a text for $L$. Let Cntrexmpls (NegSet) denote the set of all elements which are ever placed in Cntrexmpls (NegSet) by the above construction. Let $\gamma_T$ be defined as follows.

$$\gamma_T(i) = \begin{cases} ncex(\mathbf{M}(T[i], \gamma_T[i])), & \text{if } \mathbf{M}(T[i], \gamma_T[i]) \in \text{NegSet}; \\ \#, & \text{otherwise.} \end{cases}$$

For $\tau \subseteq T$, let $\gamma_\tau = \gamma_T[|\tau|]$. Note that eventually, any conjecture $j$ by $\mathbf{M}$ on $(T, \gamma_T)$ which enumerates an element not in $L$, belongs to NegSet, with a negative counterexample for it belonging to Cntrexmpls (given by $ncex(j)$). This is due to eventual success of step 3, for all $\tau \subseteq T$, for which $\mathbf{M}(\tau, \gamma_\tau) \not\subseteq L$ (due to priority assigned to step 3).

If there are infinitely many $\tau \subseteq T$ such that $\mathbf{M}(\tau, \gamma_\tau) \not\subseteq L$, then clearly, $\mathbf{M}$ does not **NCBc**-identify $L$. On the other hand, if there are only finitely many such $\tau$, then clearly all such $\tau$ would have been handled by some stage $s$, and beyond stage $s$, step 3 would never succeed. Thus, for any stage $s' \geq s$, simulation of $\mathbf{M}(\tau, \gamma_\tau)$, as at stage $s'$ step 2, is correct (i.e., negative counterexamples are given, whenever the conjectured language is not a subset of $L$), and step 2 succeeds in all but finitely many stages. Thus again infinitely many conjectures of $\mathbf{M}$ on $(T, \gamma_T)$ are incorrect (and enumerate an element of $\overline{L}$), contradicting the hypothesis.

From above cases it follows that $\mathbf{M}$ does not **NCBc**-identify $L$. Theorem follows. ∎

Our next result shows that all classes of languages learnable in our basic **Ex**-style model with arbitrary finite number of errors in almost all conjectures can be learned without errors in the basic **Bc**-style model. Note the contrast with learning from texts where $\mathbf{TxtEx}^{2j+1} - \mathbf{TxtBc}^{j} \neq \emptyset$ [CL82].

**Theorem 46** $\mathbf{NCEx}^* \subseteq \mathbf{NCBc}$.

PROOF. Suppose $\mathbf{M}$ $\mathbf{NCEx}^*$-identifies $\mathcal{L}$. Define $\mathbf{M}'$ as follows. $\mathbf{M}'$ on (positive) input $\sigma$ is obtained by simulating $\mathbf{M}$ on input $\sigma$. Suppose $\mathbf{M}$ outputs grammar $i$. If this is the first time $\mathbf{M}$ has output $i$, then $\mathbf{M}'$ also outputs $i$, and passes to $\mathbf{M}$ any negative counterexample obtained. If $i$ has been previously output by $\mathbf{M}$, then in the simulation $\mathbf{M}$ is given the negative counterexample received by $\mathbf{M}'$ the last time $i$ was output by $\mathbf{M}'$ — and then $\mathbf{M}'$ outputs a grammar for $W_i \cup \text{content}(\sigma) - \{x \mid x$ has been received by $\mathbf{M}'$ as negative counterexample upto now$\}$. Now, if the final grammar of $\mathbf{M}$ on the input text makes only finitely many errors, then all these errors are patched by $\mathbf{M}'$ (positive errors are patched due to addition of content$(\sigma)$; negative errors are patched due to fixing of all negative errors, one by one, as received by $\mathbf{M}'$). Thus, $\mathbf{M}'$ $\mathbf{NCBc}$-identifies $\mathcal{L}$. ∎

Next theorem establishes yet another limitation on the learning power of our basic **Bc**-style learning: some languages not learnable in this model can be **Bc**-learned without negative counterexamples if only one error in almost all conjectures is allowed.

**Theorem 47** $\mathbf{TxtBc}^1 - \mathbf{NCBc} \neq \emptyset$.

PROOF. Follows from $\mathbf{TxtBc}^1 - \mathbf{InfBc} \neq \emptyset$ [CL82] and Proposition 44. ∎

Now we turn to **Bc**-style learning with limited size of negative counterexamples. First, note that Theorem 26 gives us: $\mathbf{BNCEx} - \mathbf{TxtBc}^* \neq \emptyset$. In other words, some languages **Ex**-learnable with negative counterexamples of limited size cannot be **Bc**-learned without counterexamples even with an arbitrary finite number of errors in almost all conjectures. On the other hand, as the next theorem shows, some languages learnable in our basic **Ex**-style learning with negative counterexamples cannot be learned in **Bc**-model with limited size of negative counterexamples even if an arbitrary finite number of errors is allowed in almost all conjectures.

**Theorem 48** $\mathbf{NCEx} - \mathbf{BNCBc}^* \neq \emptyset$.

PROOF. The class used for separating $\mathbf{BF}^h\mathbf{NCEx} - \mathbf{BNCEx}$ in Theorem 27, INIT $\cup \{N\}$, is not in $\mathbf{BNCBc}^*$, as negative examples are not relevant and the class itself is not in $\mathbf{TxtBc}^*$ by Proposition 23. ∎

Similarly, from the proof of Theorem 28 and Corollary 29 we have,

**Theorem 49** *Suppose $h, h' \in INCFUNC$ such that $h'(x) > h(x)$ for infinitely many $x$. Then* $\mathbf{BF}^{h'}\mathbf{NCEx} - \mathbf{NRBF}^{h}\mathbf{NCBc}^{*} \neq \emptyset$.

Thus, similarly to the **Ex**-style model, we have a hierarchy on the **Bc**-style models depending on the recursive factor limiting the size of negative counterexamples.

Our next result establishes a limitation on the learning power of **Bc**-style learning with negative counterexamples of limited size allowing arbitrary finite number of errors in almost all conjectures: there are some indexed classes of languages not learnable in this model (as Theorem 14 showed, all such classes are **Ex**-style learnable in the basic model).

**Theorem 50** *There exists an indexed family not in* $\mathbf{BNCBc}^{*}$.

PROOF. The class used in the proof of Theorem 27, INIT$\cup\{N\}$, is an indexed family not in $\mathbf{BNCBc}^{*}$. ∎

**Corollary 51** $\mathbf{InfEx} - \mathbf{BNCBc}^{*} \neq \emptyset$.

Now we establish one of our most surprising results: there exists a **Bc**-style learner with negative counterexamples, allowing just one error in almost all conjectures, with the "ultimate power" - it can learn the class of all recursively enumerable languages!

**Theorem 52** $\mathcal{E} \in \mathbf{NCBc}^{1}$.

PROOF. First we give an informal idea of the proof. Our learner can clearly test if a particular $W_s \subseteq L$. Given an arbitrary initial segment of the input $T[n]$, we will want to test if content$(T[n]) \not\subseteq W_s$ for any r.e. set $W_s \subseteq L$, where $L$ is a target language. Of course, the teacher cannot directly answer such questions, since $W_s$ might not be the target language (note also that the problem is undecidable). However, the learner finds a way to encode this problem into a current conjecture and test if the current conjecture generates a subset of the target language. In order to do this, the learner potentially makes one deliberate error in its conjecture! We now proceed formally.

Define $\mathbf{M}$ on the input text $T$ as follows. Initially, it outputs a grammar for $N$. If it does not generate a negative counterexample, then we are done. Otherwise, let $c$ be the negative counterexample. Go to stage 0.

Stage $s$

1. Output grammar $s$. If it generates a negative counterexample, then go to stage $s+1$.
2. Else,
   For $n = 0$ to $\infty$ do:
      Output a grammar for the language $X$ where:

$$X = \begin{cases} \emptyset, & \text{if content}(T[n]) \not\subseteq W_s; \\ W_s \cup \{c\}, & \text{otherwise.} \end{cases}$$

      If it does not generate a negative counterexample, then go to stage $s+1$,
      Otherwise continue with the next iteration of For loop.
   EndFor
End stage $s$

We now claim that above **M** $\mathbf{NCBc}^1$-identifies $\mathcal{E}$. Clearly, if $L = N$, then **M** $\mathbf{NCBc}^1$-identifies $L$. Now suppose $L \neq N$. Let $c$ be the negative counterexample received by **M** for $N$. Let $j$ be the least grammar for $L$, and $T$ be a text for $L$. We claim that all stages $s < j$ will finish, and stage $j$ will not finish. To see this consider any $s < j$.

Case 1: $W_s \not\subseteq L$.

In this case note that step 1 would generate a negative counterexample, and thus we will go to stage $s+1$.

Case 2: Not Case 1 (i.e., $W_s \subseteq L$ but $L \not\subseteq W_s$).

In this case, let $m$ be least such that content$(T[m]) \not\subseteq W_s$. Then, in the iteration of For loop in step 2, with $n = m$, the grammar output is for $\emptyset$. Thus, there is no negative counterexample, and algorithm proceeds to stage $s+1$.

Also, note that in stage $s = j$, step 1 would not get a negative counterexample, and since $c \notin L$, every iteration of For loop will get a negative counterexample. Thus, **M** keeps outputting grammar for $W_j \cup \{c\}$. Hence **M** $\mathbf{NCBc}^1$-identifies $L$. Thus, we have that **M** $\mathbf{NCBc}^1$-identifies $\mathcal{E}$. ∎

Since $\mathcal{E} \in \mathbf{InfBc}^*$, we have

**Corollary 53** *(a)* $\mathbf{NCBc}^1 = \mathbf{InfBc}^*$.

*(b) For all* $a \in N \cup \{*\}$, $\mathbf{NCBc}^a = \mathbf{LNCBc}^a$.

The following corollary shows a contrast with respect to the case when there are no errors in conjectures (Proposition 44 and Theorem 45). What a difference just one error can make! Using the fact that $\mathbf{InfBc}^n \subsetneq \mathbf{InfBc}^*$ (see [CS83]), we get

**Corollary 54** *For all $n > 0$, $\mathbf{InfBc}^n \subsetneq \mathbf{NCBc}^n = \mathbf{NCBc}^1$.*

The ideas of the above theorem are now employed to show that all infinite recursively enumerable languages can be learned in our basic **Bc**-style model with negative counterexamples of limited size allowing just one error in almost all conjectures. Note that, as we demonstrated in Theorem 50, contrary to the case when there are no limits on the size of negative counterexamples, such learners cannot learn the class of all recursively enumerable languages.

**Theorem 55** *Let $\mathcal{L} = \{L \in \mathcal{E} \mid L$ is infinite $\}$. Then $\mathcal{L} \in \mathbf{BNCBc}^1$.*

Proof. The idea is essentially the same as showing $\mathcal{E} \in \mathbf{NCBc}^1$, (Theorem 52) except that now

(i) we need to keep conjecturing $N$ until we get negative counterexample, if any, and

(ii) we need to do step 1 check in every iteration of the For loop in step 2 (to make sure that every negative counterexample gets a chance, since **BNC** model only allows negative counterexample below the maximum element in the input).

We omit the details. ∎

As there exists a class of infinite languages which does not belong to $\mathbf{InfBc}^n$ (see [CS83]), we have

**Corollary 56** *For all $n \in N$, $\mathbf{BNCBc}^1 - \mathbf{InfBc}^n \neq \emptyset$.*

Thus, $\mathbf{BNCBc}^m$ and $\mathbf{InfBc}^n$ are incomparable for $m > 0$. The above result does not generalize to $\mathbf{InfBc}^*$, as $\mathbf{InfBc}^*$ contains the class $\mathcal{E}$.

Now, based on the ideas similar to the ones used in Theorem 52, we show that all classes of languages $\mathbf{Bc}^n$-learnable without negative counterexamples can be **Bc**-learned with negative counterexamples of limited size when one error in almost all conjectures is allowed.

**Theorem 57** *For all $n \in N$, $\mathbf{TxtBc}^n \subseteq \mathbf{BNCBc}^1$.*

Proof. Suppose $\mathbf{M}$ $\mathbf{TxtBc}^n$-identifies $\mathcal{L}$. Define $\mathbf{M}'$ as follows.

Initially, on input $T[m]$, for $m = 0, 1, 2, \ldots$, $\mathbf{M}'$ outputs grammar for the language:

$$\begin{cases} \mathrm{content}(T[m]), & \text{if } \mathrm{card}(W_{\mathbf{M}(T[m])}) \leq \mathrm{card}(\mathrm{content}(T[m])) + n; \\ N, & \text{otherwise.} \end{cases}$$

This continues until and unless a negative counterexample is received. Note that if a negative counterexample is never received then either

(i) input is a finite set, and eventually only first case above applies, and thus $\mathbf{M}'$ is outputting only correct grammars from some point onwards,

or

(ii) input is an infinite member of $\mathcal{L}$, and eventually only second case above applies, and thus input must be $N$, and $\mathbf{M}'$ is outputting only correct grammars from some point onwards,

or

(iii) input is not in $\mathcal{L}$.

So, if above process does not generate a negative counterexample, then we are done. Otherwise, let $c$ be the negative counterexample. Go to stage 0.

Stage $s$

For $m = s$ to $\infty$ do:

(Note that we start with $m = s$).

1. Output a grammar for:

$$\begin{cases} \mathrm{content}(T[m]), & \text{if } \mathrm{card}(W_{\mathbf{M}(T[m])}) \leq \mathrm{card}(\mathrm{content}(T[m])) + n; \\ W_s, & \text{otherwise.} \end{cases}$$

If it generates a negative counterexample, then go to stage $s + 1$. (note that negative counterexample can be generated only if the second case above applied).

2. Output a grammar for:

$$L = \begin{cases} \mathrm{content}(T[m]), & \text{if } \mathrm{card}(W_{\mathbf{M}(T[m])}) \\ & \leq \mathrm{card}(\mathrm{content}(T[m])) + n \\ & \text{or } \mathrm{content}(T[m]) \not\subseteq W_s; \\ \mathrm{content}(T[m]) \cup W_s \cup \{c\}, & \text{otherwise.} \end{cases}$$

If it does not generate a negative counterexample, then go to stage $s + 1$,

Otherwise continue with the next iteration of For loop.
EndFor
End stage $s$

We now claim that above $\mathbf{M}$ $\mathbf{NCBc}^1$-identifies $\mathcal{L}$. Let $L \in \mathcal{L}$ and $T$ be a text for $L$. Based on the discussion before the staging construction, assume that we reach stage 0 (otherwise we are already done).

Now, suppose $L$ is finite and $T$ is a text for $L$. Then, for some $t$, for all $m \geq t$, content($T[m]$) $= L$ and $\mathbf{M}(T[m])$ outputs a grammar for an $n$-variant of $L$. Thus, irrespective of whether we converge to a stage or have infinitely many stages, eventually only grammar for $L$ would be output by $\mathbf{M}'$, as first clause applies for grammars output in steps 1 and 2, for all stages $m \geq t$ in the staging construction.

So suppose $L$ is infinite and $T$ is a text for $L$. Then for some $t$, for all $m \geq t$, $\mathbf{M}(T[m])$ is a grammar for infinite set. Now note that, any stage $j$ for which $W_j \neq L$ would be exited (same argument as done in the proof of Theorem 52 for $\mathcal{E} \in \mathbf{NCBc}^1$ applies here). Furthermore, for any $j \geq t$, such that $j$ is a grammar for $L$, stage $j$ would not be exited. (We may exit some stages $j < t$, for which $W_j$ is a grammar for $L$, due to "card($W_{\mathbf{M}(T[m])}$) $\leq$ card(content($T[m]$)) $+ n$" part in clause 1 of step 2). Thus, eventually we reach a stage $s = j$ such that $W_j$ is a grammar for $L$, and the construction does not leave stage $s$. From this point onwards, $\mathbf{M}$ only outputs a grammar for $W_s$ or for $W_s \cup \{c\}$. Theorem follows. ∎

Similarly, one can show

**Theorem 58** $\mathbf{TxtEx}^* \subseteq \mathbf{BNCBc}^1$.

Similarly to the case of **Ex**-style learning, **BNCBc** and **NCBc** models turn out to be equivalent for the classes of infinite languages.

**Theorem 59** *Suppose $\mathcal{L}$ consists of only infinite languages. Then $\mathcal{L} \in$* **NCBc** *iff $\mathcal{L} \in$* **BNCBc**.

Proof. The idea of the proof is similar to the proof of Theorem 25. The main difference being that we do not patch errors, and the argument about eventually being able to give right answers in the simulation being based on "finitely many wrong conjectures done by **NCBc**-learner", rather than "finitely many conjectures by **NCEx**-learner." We now proceed formally.

Clearly, if $\mathcal{L} \in$ **BNCBc**, then $\mathcal{L} \in$ **NCBc**. So we only need to show that if $\mathcal{L} \in$ **NCBc** then $\mathcal{L} \in$ **BNCBc**. Suppose $\mathbf{M}$ **NCBc**-identifies $\mathcal{L}$. Define $\mathbf{M}'$ as follows. $\mathbf{M}'$ on the input text $T$ of positive data for an infinite language $L$

behaves as follows. Initially let NegSet $= \emptyset$. Intuitively, NegSet denotes the set of grammars for which we know a negative counterexample. For $j \in$ NegSet, $ncex(j)$ would denote a negative counterexample for $j$. For ease of presentation, we will let $\mathbf{M}'$ output more than one conjecture (one after another) at some input point and get negative counterexamples for each of them. This is for ease of presentation and one can always spread out the conjectures.

Stage $s$ of $\mathbf{M}'$ (after seeing input $T[s]$)

1. Simulate $\mathbf{M}$ on $T[s]$, by giving negative counterexamples to any conjectures $j \in$ NegSet by $ncex(j)$. Other grammars get # as counterexample.
2. Let $S$ be the set of conjectures output by $\mathbf{M}$ on initial segments of $T[s]$.
3. Output a grammar for $\bigcup_{i \in S - \mathrm{NegSet}} W_i$.
4. If there is no negative counterexample, then go to stage $s+1$.
5. Else (i.e., there is a negative counterexample), output one by one, elements of $S -$ NegSet. For each $i \in S -$ NegSet, if a negative counterexample is obtained, then place $i$ in NegSet and define $ncex(i)$ to be this negative counterexample.
6. Go to stage $s+1$.

End Stage $s$.

Now suppose $T$ is a text for $L \in \mathcal{L}$ and consider the above construction for $\mathbf{M}'$. Due to output at step 3, eventually any grammar output by $\mathbf{M}$ on text $T$ (when negative counterexamples are based on NegSet and $ncex$), which enumerates a non-subset of $L$ would receive a negative counterexample. Thus, as $\mathbf{M}$ $\mathbf{NCBc}$-identifies $L$, for all but finitely many stages $s$, all the answers given to $\mathbf{M}$ in step 1 would be correct. Thus, the grammar output in step 3 would be correct for all but finitely many stages, and $\mathbf{M}'$ $\mathbf{BNCBc}$-identifies $L$. ∎

We now mention some of the open questions regarding behaviourally correct learning when the size of the negative counterexamples is bounded.

Open Question: Is the $\mathbf{BNCBc}^n$ hierarchy strict?

Open Question: Is $\mathbf{TxtBc}^* \subseteq \mathbf{BNCBc}^1$?

## 7 Conclusions

In this paper we introduced three different models of learning with counterexamples, and studied their relationship with other known criteria. Among the interesting results we showed, there are the result that $\mathcal{E}$ can be learned in

**NCBc**[1], and the result that **InfEx** − **NCEx** $\neq \emptyset$, despite the fact that a **NCEx**-learner can determine the membership for any particular element.

The model of (**Ex**-) learning with counterexamples was shown to be quite robust as various modifications of the model which allow delays in the counterexample, or even the model in which the learner is informed about the existence of the counterexample but not given any specific counterexample, turn out to be of same learning power.

We now remark on some of the complexity advantages of having negative counterexamples. Note that the class $\mathcal{L}_1 = \{L \mid \mathrm{card}(N-L) = 1\}$ is in **TxtEx**, but requires unbounded number of mind changes to learn. On the other hand, $\mathcal{L}_1$ can be easily learned using one mind change if negative counterexamples are available (learner can first output a grammar for $N$; the counterexample then gives away the language being learned (assuming it is from the class)). Thus, not only does **NCEx** model give learnability advantages over **TxtEx**, it also gives complexity advantages over **TxtEx** for some classes in **TxtEx**. Note here that if one does not allow any mind changes, then **NCEx** and **TxtEx** are both the same; thus the above result is the best mind change complexity advantage possible. Simlarly, if we consider the class $\mathcal{L}_2 = \mathrm{INIT} \cup \{N\}$, then it is learnable in **NCEx** model, but the number of mind changes is unbounded. However, $\mathcal{L}_2$ can be learned by using at most one mind change in the model **LNCEx** (by a learner which outputs a grammar for $N$ initially; least counterexample, if any, would then determine the input language) [3]. Thus, even though **LNCEx** does not give learnability advantages over **NCEx**, it does give complexity advantages.

Let $a \dot{-} b = a - b$, if $a \geq b$; $a \dot{-} b = 0$ otherwise; Consider the class:
$\mathcal{L}_3 = \{L \mid (\exists! e)[\langle 0, e\rangle \in L \;\;\&\;\; L - \{\langle 0, e\rangle\} \subseteq \{\langle x, y\rangle \mid x > 1\} \;\;\&\;\; \mathrm{card}(L - \{\langle 0, e\rangle\}) = e \dot{-} \min(W_e)]\}$

$\mathcal{L}_3$ is in **LNCEx** with at most one mind change (an **LNCEx**-learner first looks for $\langle 0, e\rangle$ in the input, and then outputs a grammar for $\{\langle 0, x\rangle \mid x \leq e, x \in W_e\}$; now the least counterexample allows the learner to calculate the minimal element in $W_e$ and hence the cardinality of the input language — which is enough to identify the input language using one mind change). However $\mathcal{L}_3$ cannot be learned in **InfEx** using bounded number of mind changes. Note that **LNCEx** $\subset$ **InfEx**. So getting negative counterexamples gives complexity advantages over informants, despite informant being more advantageous for learning as a whole.

[3] If we want to consider a language class within **TxtEx** to show the complexity advantage of **LNCEx** over **NCEx**, then one can choose the class as follows. Let $L_i = \{\langle i, x\rangle \mid x \in N\}$. Let $L_i^n = \{\langle i, x\rangle \mid x \leq n\}$. Let $\mathcal{L}_i = \{L_i\} \cup \{L_i^n \mid n \leq i\}$. Then $\mathcal{L} = \bigcup_i \mathcal{L}_i$ is in **TxtEx**, cannot be learned with bounded number of mind changes in **NCEx**, however can be learned with one mind change in **LNCEx**.

The situation is more complex in considering the complexity advantages of **NCEx**-model compared to **InfEx** model. Consider the following classes. Let

$$L_i = \{\langle i, x\rangle \mid x \in N\}$$

$$\mathcal{L}_4 = \{L_0\} \cup \{L \mid (\exists x > 0)[L = L_0 \cup L_x]\}$$

$$\mathcal{L}_5 = \{L \mid (\exists x > 0)(\exists y)[L = (L_0 - \langle 0, \langle x, y\rangle\rangle) \cup L_x]\}$$

It is easy to verify that one can identify $\mathcal{L}_4 \cup \mathcal{L}_5$ using one mind change in the **NCEx** model (learner can first output a grammar for $L_0$; then presence or absence of counterexample determines whether the input language is from $\mathcal{L}_4$ or $\mathcal{L}_5$, which can then be identified using (possibly) one more mind change). On the other hand it can be shown that $\mathcal{L}_4 \cup \mathcal{L}_5$ cannot be **InfEx**-identified with at most one mind change (it needs 2 mind changes). A generalization of the above class can be used to show that there exist classes which can be **NCEx**-identifies using $n-1$ mind changes, but cannot be **InfEx**-identified using $(2^n - 1) - 2$ mind changes. This is optimal as it can be shown that any class which can be **NCEx**-identified using $n-1$ mind changes can also be identified using $(2^n - 1) - 1$ mind changes in **InfEx**-model. We omit the details.

## 8 Acknowledgements

We would like to thank anonymous referees for helpful suggestions. We dedicate this paper to the memory of our dear friend and colleague Carl Smith.

## References

[Ang88] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[Bār74] J. Bārzdiņš. Two theorems on the limiting synthesis of functions. In *Theory of Algorithms and Programs, vol. 1*, pages 82–88. Latvian State University, 1974. In Russian.

[BB75] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[BCJ95] G. Baliga, J. Case, and S. Jain. Language learning with some negative information. *Journal of Computer and System Sciences*, 51(5):273–285, 1995.

[BH70] R. Brown and C. Hanlon. Derivational complexity and the order of acquisition in child speech. In J. R. Hayes, editor, *Cognition and the Development of Language*. Wiley, 1970.

[Blu67] M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.

[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.

[CS83] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[DPS86] M. Demetras, K. Post, and C. Snow. Feedback to first language learners: The role of repetitions and clarification questions. *Journal of Child Language*, 13:275–292, 1986.

[FGJ+94] L. Fortnow, W. Gasarch, S. Jain, E. Kinber, M. Kummer, S. Kurtz, M. Pleszkoch, T. Slaman, R. Solovay, and F. Stephan. Extremes in the degrees of inferability. *Annals of Pure and Applied Logic*, 66:231–276, 1994.

[GM98] W. Gasarch and G. Martin. *Bounded Queries in Recursion Theory.* Birkhauser, 1998.

[Gol67] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[GP89] W. Gasarch and M. Pleszkoch. Learning via queries to an oracle. In R. Rivest, D. Haussler, and M. Warmuth, editors, *Proceedings of the Second Annual Workshop on Computational Learning Theory*, pages 214–229. Morgan Kaufmann, 1989.

[HPTS84] K. Hirsh-Pasek, R. Treiman, and M. Schneiderman. Brown and Hanlon revisited: Mothers' sensitivity to ungrammatical forms. *Journal of Child Language*, 11:81–88, 1984.

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[LNZ02] S. Lange, J. Nessel, and S. Zilles. Learning languages with queries. In *Proceedings of Treffen der GI-Fachgruppe Maschinelles Lernen (FGML), Learning Lab Lower Saxony, Hannover, Germany*, pages 92–99, 2002.

[LZ94] S. Lange and T. Zeugmann. Characterization of language learning from informant under various monotonicity constraints. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:73–94, 1994.

[LZ04a] S. Lange and S. Zilles. Comparison of query learning and gold-style learning in dependence of the hypothesis space. In John Case Shai Ben-David and Akira Maruoka, editors, *Algorithmic Learning Theory: Fifteenth International Conference (ALT' 2004)*, volume 3244 of *Lecture Notes in Artificial Intelligence*, pages 99–113. Springer-Verlag, 2004.

[LZ04b] S. Lange and S. Zilles. Replacing limit learners with equally powerful one-shot query learners. In John Shawe-Taylor and Yoram Singer, editors, *Proceedings of the Seventeenth Annual Conference on Computational Learning Theory*, volume 3120 of *Lecture Notes in Artificial Intelligence*, pages 155–169. Springer-Verlag, 2004.

[Mot91] T. Motoki. Inductive inference from all positive and some negative data. *Information Processing Letters*, 39(4):177–182, 1991.

[MY78] M. Machtey and P. Young. *An Introduction to the General Theory of Algorithms*. North Holland, New York, 1978.

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

[Shi86] T. Shinohara. *Studies on Inductive Inference from Positive Data*. PhD thesis, Kyushu University, Kyushu, Japan, 1986.