1-2006

Learning Languages from Positive Data and a Finite Number of Queries

Sanjay Jain
National University of Singapore

Efim Kinber
Sacred Heart University, kinbere@sacredheart.edu

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information and Computation 204 (2006) 123–175
Information and Computation
www.elsevier.com/locate/ic

Learning languages from positive data and a finite number of queries

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

[a] *School of Computing, National University of Singapore, Singapore 117543, Singapore*
[b] *Department of Computer Science, Sacred Heart University, Fairfield, CT 06432-1000, USA*

Received 10 August 2005; revised 12 September 2005
Available online 8 November 2005

Abstract

A computational model for learning languages in the limit from full positive data and a bounded number of queries to the teacher (oracle) is introduced and explored. Equivalence, superset, and subset queries are considered (for the latter one we consider also a variant when the learner tests every conjecture, but the number of negative answers is uniformly bounded). If the answer is negative, the teacher may provide a counterexample. We consider several types of counterexamples: arbitrary, least counterexamples, the ones whose size is bounded by the size of positive data seen so far, and no counterexamples. A number of hierarchies based on the number of queries (answers) and types of answers/ counterexamples is established. Capabilities of learning with different types of queries are compared. In most cases, one or two queries of one type can sometimes do more than any bounded number of queries of another type. Still, surprisingly, a finite number of subset queries is sufficient to simulate the same number of equivalence queries when *behaviourally correct* learners do not receive counterexamples and may have unbounded number of errors in almost all conjectures.

* Corresponding author. Fax: +65 6779 4580.
E-mail addresses: sanjay@comp.nus.edu.sg (S. Jain), kinbere@sacredheart.edu (E. Kinber).
[1] This work was supported in part by NUS Grant No. R252-000-127-112.

0890-5401/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2005.09.001

1. Introduction

Finding an adequate computational model for learning languages has been an important objective for last four decades. In 1967, Gold [15] introduced a classical model of learning languages in the limit from full positive data (that is, all correct statements in the target language). Under Gold's paradigm, the learner stabilizes to a correct grammar of the target language (**Ex**-style learning). Based on the same idea of learning in the limit, Case and Lynes [10] and Osherson and Weinstein [27] (see also [6,11]) introduced a more powerful *behaviorally correct* type of learning languages, when a learner almost always outputs correct (but not necessarily the same) grammars for the target language (**Bc**-style learning). In both cases, the authors also considered a much stronger (and less realistic) model of learning languages in the presence of full positive and *negative* data. In [8] the authors considered an intermediate model, where a learner gets full positive data and a finite number of negative examples. However, negative data in the latter paper is preselected, and, thus, dramatically affects learning capabilities.

In the paper [4], D. Angluin introduced another important learning paradigm, i.e. learning from queries to a teacher (oracle). Among others, D. Angluin introduced three types of queries: equivalence queries—when a learner asks if the current conjecture generates the target language; subset and superset queries—when a learner asks if the current conjecture generates a subset or a superset of the target language, respectively. If the answer is negative, the teacher may provide a *counterexample* showing where the current conjecture errs. This learning paradigm of testing conjectures against the target concept (and some other related types of queries) has been explored, primarily in the context of learning finite concepts and regular languages, in several papers, for example, [5,25,3,1,21,29,18]. In [22], the authors applied this paradigm to explore learning (potentially infinite) languages without knowing any data in advance (neither positive, nor negative) (see also [24,23]). A somewhat different types of queries (where one may ask queries to an oracle such as halting problem) was considered in [17,16,12].

In this paper, we combine learning languages from positive data and learning languages from queries into one model. On one hand, this model reflects the fact that a child, during a process of acquisition of a new language, potentially gets access to all correct statements. On the other hand, this model provides an important tool available to a child: a possibility to communicate with a teacher testing conjectures about the grammar describing the target language. The first attempt of combining the abovementioned paradigms of learning was made in [19], where learning from positive data and negative counterexamples to conjectures was considered. In this model, a learner essentially asks a subset query about every conjecture. Thus, a learner, being provided with full positive data, is concerned with " overgeneralizing," that is, including into conjectures data not belonging to the target language. If the current conjecture is not a subset, the teacher may provide a negative counterexample. In the sequel, we will refer to the model defined in [19], as learning using negative counterexamples to conjectures. In the current paper, we concentrate on the case when a learner can query the teacher only a bounded (finite) number of times - thus, limiting the amount of help from the teacher. As avoiding overgeneralization is probably the main challenge a language learner can face (see, for example [26,31]), exploring help from subset queries is our primary objective in this paper. In addition to subset queries, we also consider learning with equivalence and superset queries. Using the latter type of queries in the presence of full positive data may seem problematic, as " counterexamples" in this case are positive, and the learner gets them eventually

anyway. However, sometimes, a teacher may have difficulty providing negative counterexamples. Moreover, as we have shown, positive counterexamples can help learning language that cannot be learned otherwise – even when full positive data is eventually available!

We also consider the model of learning using negative counterexamples to conjectures as defined in [19] — when the number of (negative) counterexamples is uniformly bounded. On the surface, this type of learning seems to be at least as capable as learning with a bounded number of subset queries (recall that in the former model, the learner asks a subset query about its conjectures). However, as we have shown, surprisingly, there exist classes of languages learnable with just one subset query, but not learnable receiving any bounded number of negative counterexamples to conjectures!

As the number of queries in our learning model is always uniformly bounded, it can naturally be considered as a measure of complexity of learning languages (number of queries as a measure of complexity of solving hard computational problems has been extensively explored, see, for example [14]).

Following [19], in addition to the case when counterexamples provided by the teacher are arbitrary (our basic learning model), we consider three variants of this basic model:

- the learner always gets the least counterexample (Ibarra and Jiang [18] explored this type of learning using equivalence queries for finite deterministic automata);
- the counterexample is bounded by the largest positive data seen so far;
- the learner gets only answers "yes" or "no," but no counterexamples (queries of this type are known as *restricted*).

The latter two variants address complexity issues: a teacher might not be able to compute a long counterexample in a reasonable time, or might not be able to provide it at all.[2]

In this paper we explore effects of different types of queries on learning capabilities. In particular, we explore:

- how the number of queries can affect learning capabilities (hierarchies based on the number of queries);
- relationships between learning capabilities based on different types of queries;
- how three different variants of the basic model (described above) using different types of counterexamples given affect learning capabilities;
- the relationship between learning using subset queries and learning using negative counterexamples to conjectures; even though, for **Ex**-type learning, these models coincide when unbounded finite number of subset queries is allowed, some subtle differences arise when one bounds the number of queries or counterexamples provided.

[2] The teacher must be able to solve the subset, equivalence, and superset problems for recursively enumerable sets. These problems are algorithmically unsolvable in the general case. However, in many of the examples considered in this paper, these problems are solvable. Moreover, exploring computability and learnability using oracles proved to be very helpful for better understanding of nature and capabilities of both in various contexts even when algorithmic solvability would be problematic [28,16,12,22].

The paper is organized as follows. Section 2 is devoted to notation and some basic definitions (in particular, definitions of **Ex** and **Bc** types of learning). In Section 3, we define learning from positive data via subset, equivalence, and superset queries, as well as three abovementioned variants of the basic learning model. We also show here that learning with counterexamples bounded by the largest positive data seen so far does not help for all three types of queries—even if the finite number of queries is not uniformly bounded. In Section 4 we define learning with a bounded number of negative counterexamples to conjectures.

In Section 5 general hierarchies based on the number of queries are exhibited. Our results here (Theorems 19 and 22) show that, for all three types of queries, learning with $(n+1)$ queries is stronger than with n queries. Moreover, classes of languages witnessing hierarchies in question can be **Ex**-learned using $(n+1)$ restricted queries (providing only answers "yes" or "no"), but cannot be learned by **Bc**-type learners getting the least counterexamples.

In Section 6, we establish hierarchies based on the differences between different variants of the basic learning model: using least counterexamples versus arbitrary counterexamples, and arbitrary counterexamples versus no counterexamples. First, we show that, for all three types of queries, when only one query is permitted, getting the least counterexample helps no better than getting no counterexample (Theorem 25). On the other hand, (again for all three types of queries) **Ex**-learners making just two queries and receiving the least counterexamples can do better than **Bc**-learners making n queries, making a finite number of errors in almost all conjectures, and receiving arbitrary counterexamples to queries (Theorems 26 and 29). Interestingly, one and the same class witnesses this hierarchy for both subset and equivalence types of queries. A somewhat surprising hierarchy has been found for the case of learning with bounded number of negative counterexamples to conjectures: learners getting $(2n-1)$ arbitrary negative counterexamples to conjectures can learn at least as much as the ones getting n least negative counterexamples, and the bound $(2n-1)$ is tight—$(2n-2)$ arbitrary examples are not enough to simulate n least negative counterexamples (Theorems 32 and 33). In the rest of the section we demonstrate that **Ex**-learners making just two queries and getting arbitrary counterexamples can learn classes not **Bc**-learnable via any n queries with no counterexamples, even when a finite number of errors is allowed in almost all conjectures (Theorems 36 and 37). Again, the hierarchies for subset and equivalence queries are witnessed by the same class of languages.

In Section 7, we exhibit subtle differences between learning via bounded number of subset queries and learning with bounded number of counterexamples to conjectures. Our main, quite surprising result in this section (Theorem 43) shows that **Ex**-learners making just one subset query with no counterexample can learn some class of languages that is not learnable by **Bc**-learners which are provided with at most n (least) counterexamples to their conjectures, even if allowed any finite number of errors in almost all conjectures! (The class of languages witnessing this result can also be learned via one restricted equivalence query). On the other hand, **Ex**-learners which are provided with one negative counterexample to their conjectures (if the counterexample exists), can learn some class which is not learnable by **Bc**-learners making (at most) n subset queries and allowing any bounded number of errors in almost all conjectures (Theorem 47; Theorem 50 also exhibits a slightly different version of the above phenomenon).

In Section 8, we explore how finite number of subset queries (including learning with a bounded number of negative counterexamples) helps to learn compared with finite number of other types of queries. We show that there are classes of languages **Ex**-learnable with one restricted subset

query (or with at most one negative counterexample to their conjectures) but not **Bc**-learnable with any finite number of equivalence queries, even when always getting least counterexamples and allowing any finite number of errors in almost all conjectures (Theorem 56). In Section 9, we explore how finite number of equivalence or superset queries fairs against a finite number of subset queries (or a bounded number of negative counterexamples to conjectures). First, we show that **Ex**-learners using just one restricted superset or equivalence query can learn a class not learnable by **Bc**-learners which are given negative counterexamples (if applicable) to all its conjectures (Theorem 59). Then we use this result to demonstrate that **Ex**-learners making just one restricted equivalence or superset query can sometimes do better than **Bc**-learners making n subset queries or getting (at most) n least negative counterexamples to its conjectures, when a bounded finite number of errors is allowed in almost all conjectures (Corollary 61). We also discovered a subtle difference with the above result in the case when **Bc**-learners can make any unbounded finite number of errors in almost all conjectures: in this case, **Bc**-learners using n restricted equivalence queries cannot learn more than **Bc**-learners using the same number of restricted subset queries (Theorem 62). Still, if the teacher provides counterexamples, **Ex**-learners making just two equivalence queries can do better than **Bc**-learners making any finite (unbounded) number of subset queries, getting least counterexamples and making any finite (unbounded) number of errors in almost all conjectures (Theorem 63). In Section 10, we prove just one result (Theorem 66) showing that **Ex**-learners making just one restricted superset query can do better than **Bc**-learners making n equivalence queries, getting least counterexamples, and making finite (bounded) number of errors in almost all conjectures. In Section 11 we consider anomaly hierarchy.

2. Notation and preliminaries

Any unexplained recursion theoretic notation is from [28]. The symbol N denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$. Symbols $\emptyset$, $\subseteq$, $\subset$, $\supseteq$, and $\supset$ denote empty set, subset, proper subset, superset, and proper superset, respectively. $D_0, D_1, \ldots$, denotes a canonical recursive indexing of all the finite sets [28, p. 70]. We assume that if $D_i \subseteq D_j$ then $i \leqslant j$ (the canonical indexing defined in [28] satisfies this property). 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 $\min(\emptyset) = \infty$. $L_1 \mathbf{\Delta} L_2$ denotes the symmetric difference of L_1 and L_2, that is $L_1 \mathbf{\Delta} L_2 = (L_1 - L_2) \cup (L_2 - L_1)$. For a natural number a, we say that $L_1 =^a L_2$, iff $\mathrm{card}(L_1 \mathbf{\Delta} L_2) \leqslant a$. We say that $L_1 =^* L_2$, iff $\mathrm{card}(L_1 \mathbf{\Delta} L_2) < \infty$. 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 [28]. 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 $\pi_2(\langle x, y \rangle) = y$. We can extend pairing function to multiple arguments by using $\langle i_1, i_2, \ldots, i_k \rangle = \langle i_1, \langle i_2, \langle \ldots, \langle i_{k-1}, i_k \rangle \rangle \rangle \rangle$.

We let $\{W_i\}_{i \in N}$ denote an acceptable numbering of all r.e. sets. Symbol $\mathcal{E}$ will denote the set of all r.e.languages. Symbol L, with or without decorations, ranges over $\mathcal{E}$. 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 W_i enumerated within s steps, in some standard method of enumerating W_i.

We let $K = \{i \mid i \in W_i\}$. Note that K is a recursively enumerable but not recursive set [28].

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

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

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

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

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

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

Definition 2 ([15])**.** (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 ([15])**.** A *language learning machine from texts* is an algorithmic device which computes a mapping from SEQ into N.

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

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 4 ([15,10])**.** Suppose $a \in N \cup \{*\}$.

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

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

(c) $\mathbf{M}$ **TxtEx**^{a}*-identifies a class* $\mathcal{L}$ of r.e. languages (written: $\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})$) just in case $\mathbf{M}$ **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 5 ([10])**.** Suppose $a \in N \cup \{*\}$.

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

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

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

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

For $a = 0$, we often write **TxtEx** and **TxtBc**, instead of $\mathbf{TxtEx}^0$ and $\mathbf{TxtBc}^0$, respectively.

Definition 6 ([13]). σ is said to be an **TxtEx**-*stabilizing sequence* for $\mathbf{M}$ on L, iff (i) $\text{content}(\sigma) \subseteq L$, and (ii) for all σ' such that $\sigma \subseteq \sigma'$ and $\text{content}(\sigma') \subseteq L$, $\mathbf{M}(\sigma) = \mathbf{M}(\sigma')$.

Definition 7 ([7,26]). For $a \in N \cup \{*\}$, σ is said to be an $\mathbf{TxtEx}^a$-*locking sequence* for $\mathbf{M}$ on L, iff (i) $\text{content}(\sigma) \subseteq L$, (ii) for all σ' such that $\sigma \subseteq \sigma'$ and $\text{content}(\sigma') \subseteq L$, $\mathbf{M}(\sigma) = \mathbf{M}(\sigma')$, and (iii) $W_{\mathbf{M}(\sigma)} =^a L$.

Theorem 8 ([7,26]). *Suppose* $\mathbf{M}$ $\mathbf{TxtEx}^a$*-identifies* $\mathcal{L}$. *Then, there exists an* $\mathbf{TxtEx}^a$*-locking sequence for* $\mathbf{M}$ *on* L.

Definition 9 (Based on [7,26]). For $a \in N \cup \{*\}$, σ is said to be an $\mathbf{TxtBc}^a$-*locking sequence* for $\mathbf{M}$ on L, iff (i) $\text{content}(\sigma) \subseteq L$, and (ii) for all σ' such that $\sigma \subseteq \sigma'$ and $\text{content}(\sigma') \subseteq L$, $W_{\mathbf{M}(\sigma')} =^a L$.

Theorem 10 (Based on [7,26]). *Suppose* $\mathbf{M}$ $\mathbf{TxtBc}^a$*-identifies* $\mathcal{L}$. *Then, there exists a* $\mathbf{TxtBc}^a$*-locking sequence for* $\mathbf{M}$ *on* L.

Similar stabilizing sequence/locking sequence results can be obtained for criteria of inference discussed below.

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

For any L, let $cyl(L) = \{\langle i, x\rangle \mid i \in L, x \in N\}$. Let $cyl(\mathcal{L}) = \{cyl(L) \mid L \in \mathcal{L}\}$.

Let CYL_i denote the language $\{\langle i, x\rangle \mid x \in N\}$.

Let **FINITE** denote the class of all finite languages.

The following propositions are useful in proving many of our results.

Proposition 11 ([15]). *Suppose* L *is an infinite language,* $S \subseteq L$, *and* $L - S$ *is infinite. Let* $C_0 \subseteq C_1 \subseteq \cdots$ *be an infinite sequence of finite sets such that* $\bigcup_i C_i = L$. *Then* $\{L\} \cup \{S \cup C_i \mid i \in N\}$ *is not in* $\mathbf{TxtBc}^*$.

Proposition 12. *Suppose* L *is infinite and* $R_1, R_2, \ldots$ *are infinitely many pairwise disjoint subsets of* L, *where each* R_i *is infinite. Then,* $\mathcal{L} = \{X \mid X = L \text{ or } (\exists i)[X = L - R_i]\} \notin \mathbf{TxtBc}^*$.

Proof. Suppose by way of contradiction that $\mathbf{M}$ witnesses that $\{X \mid X = L \text{ or } (\exists i)[X = L - R_i]\} \in \mathbf{TxtBc}^*$. Then, let σ be a $\mathbf{TxtBc}^*$-locking sequence for $\mathbf{M}$ on L. Now for all $\sigma' \supseteq \sigma$, such that $\text{content}(\sigma') \subseteq L$, we must have that $W_{\mathbf{M}(\sigma')} =^* L$. But then $\mathbf{M}$ cannot $\mathbf{TxtBc}^*$-identify any language X such that $\text{content}(\sigma) \subseteq X \subseteq L$, and $L - X$ is infinite. Let i be such that R_i does not intersect with $\text{content}(\sigma)$. Choosing $X = L - R_i$, now shows that $\mathbf{M}$ cannot $\mathbf{TxtBc}^*$-identify $\mathcal{L}$. □

3. Learning with queries

In this section we define learning with queries. The kind of queries considered are

(i) subset queries, i.e., for a queried language Q, "is $Q \subseteq L$?," where L is the language being learned;
(ii) equivalence queries, i.e., for a queried language Q, "is $Q = L$?," where L is the language being learned;

(iii) superset queries, i.e., for a queried language Q, "is $Q \supseteq L$?," where L is the language being learned.

In the model of learning, the learner is allowed to ask queries such as above during its computation. If the answer to query is "no," we additionally can have the following possibilities:

(a) Learner is given an arbitrary counterexample (for subset query, counterexample is a member of $Q - L$; for equivalence query the counterexample is a member of $L\Delta Q$; for superset query the counterexample is a member of $L - Q$);
(b) Learner is given the least counterexample;
(c) Learner is just given the answer 'no,' without any counterexample.

We would often also consider bounds on the number of queries. We first formalize the definition of a learner which uses queries.

Definition 13. A learner using queries, can ask a query of form "$W_j \subseteq L$?" ("$W_j = L$?," "$W_j \supseteq L$?") on any input σ. Answer to the query is "yes" or "no" (along with a possible counterexample). Then, based on input σ and answers received for queries made on prefixes of σ, $\mathbf{M}$ outputs a conjecture (from N).

We assume without loss of generality that on any particular input σ, $\mathbf{M}$ asks at most one query. Also note that the queries we allow are for recursively enumerable languages, which are posed to the teacher using a grammar (index) for the language. Many of our diagonalization results (though not all) would still stand even if one uses arbitrary type of query language. However simulation results crucially use the queries being made only via grammars for the queried languages.

We now formalize learning via subset queries.

Definition 14. Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{SubQ}^a\mathbf{Ex}$-*identifies* a language L (written: $L \in \mathbf{SubQ}^a\mathbf{Ex}(\mathbf{M})$) iff for any text T for L, it behaves as follows:
  (i) The number of queries $\mathbf{M}$ asks on prefixes of T is bounded by a (if $a = *$, then the number of such queries is finite). Furthermore, all the queries are of the form "$W_j \subseteq L$?"
  (ii) Suppose the answers to the queries are made as follows. For a query "$W_j \subseteq L$?," the answer is "yes" if $W_j \subseteq L$, and the answer is "no" if $W_j - L \neq \emptyset$. For "no" answers, $\mathbf{M}$ is also provided with a counterexample, $x \in W_j - L$. Then, for some k such that $W_k = L$, for all but finitely many n, $\mathbf{M}(T[n])$ outputs the grammar k.
(b) $\mathbf{M}$ $\mathbf{SubQ}^a\mathbf{Ex}$-*identifies* a class $\mathcal{L}$ of languages (written: $\mathcal{L} \subseteq \mathbf{SubQ}^a\mathbf{Ex}(\mathbf{M})$) iff it $\mathbf{SubQ}^a\mathbf{Ex}$-identifies each $L \in \mathcal{L}$.
(c) $\mathbf{SubQ}^a\mathbf{Ex} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{SubQ}^a\mathbf{Ex}(\mathbf{M})]\}$.

$\mathbf{LSubQ}^a\mathbf{Ex}$-identification and $\mathbf{ResSubQ}^a\mathbf{Ex}$-identification can be defined similarly, where for $\mathbf{LSubQ}^a\mathbf{Ex}$-identification the learner gets the least counterexample for "no" answers, and for $\mathbf{ResSubQ}^a\mathbf{Ex}$-identification, the learner does not get any counterexample along with the "no" answers.

Now we define the variant of learning with subset queries where queries and, respectively, answers are only based on the elements bounded by the largest positive element seen so far. We call such queries *bounded* queries.

Definition 15. Suppose $a \in N \cup \{*\}$.

(a) **M BSubQ**^{a}**Ex**-*identifies* a language L (written: $\mathcal{L} \in \mathbf{BSubQ}^a\mathbf{Ex}(\mathbf{M})$) iff for any text T for L, it behaves as follows:
  (i) The number of prefixes of T on which **M** asks a query is bounded by a (if $a = *$, then the number of such prefixes of T is finite). Furthermore, all the queries are of the form "$W_j \subseteq L$?"
  (ii) Suppose the answers to the queries are made as follows. For a query "$W_j \subseteq L$?" on input $T[m]$, the answer is "yes" if $W_j \cap \{x \mid x \leqslant \max(\text{content}(T[m]))\} \subseteq L$, and the answer is "no" if $W_j \cap \{x \mid x \leqslant \max(\text{content}(T[m]))\} - L \neq \emptyset$. For "no" answers, **M** is also provided with a counterexample, $x \in W_j \cap \{x \mid x \leqslant \max(\text{content}(T[m]))\} - L$. Then, for some k such that $W_k = L$, for all but finitely many n, $\mathbf{M}(T[n])$ outputs the grammar k.

(b) **M BSubQ**^{a}**Ex**-*identifies* a class $\mathcal{L}$ of languages (written: $\mathcal{L} \subseteq \mathbf{BSubQ}^a\mathbf{Ex}(\mathbf{M})$) iff it **BSubQ**^{a}**Ex**-identifies each $L \in \mathcal{L}$.

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

For $a, b \in N \cup \{*\}$, for $\mathbf{I} \in \{\mathbf{Ex}^b, \mathbf{Bc}^b\}$, one can similarly define **SubQ**^{a}**I**, **SupQ**^{a}**I**, **EquQ**^{a}**I**, **LSubQ**^{a}**I**, **LSupQ**^{a}**I**, **LEquQ**^{a}**I**, **ResSubQ**^{a}**I**, **ResSupQ**^{a}**I**, **ResEquQ**^{a}**I**, **BSubQ**^{a}**I**, **BSupQ**^{a}**I**, and **BEquQ**^{a}**I**.

For identification with queries, where there is a bound n on the number of queries asked, we will assume without loss of generality that the learner never asks more than n queries, irrespective of whether the input language belongs to the class being learned, or whether the answers given to earlier queries are correct.

The following theorem shows that bounded queries are not useful. Thus, we will not deal with bounded counterexamples to queries from now on (note that bounded counterexamples for **NC**-type learning (defined formally in Section 4 below) are useful. Thus we will continue to use them in the context of **NC**-learning).

Theorem 16. *Suppose* $a \in N \cup \{*\}$, $n \in N$, $\mathbf{I} \in \{\mathbf{Ex}^a, \mathbf{Bc}^a\}$.

(*a*) $\mathbf{BSubQ}^*\mathbf{I} = \mathbf{TxtI}$.
(*b*) $\mathbf{BEquQ}^*\mathbf{I} = \mathbf{TxtI}$.
(*c*) $\mathbf{BSupQ}^*\mathbf{I} = \mathbf{TxtI}$.

Proof. (a) Since $\mathbf{TxtI} \subseteq \mathbf{BSubQ}^*\mathbf{I}$, it suffices to show that $\mathbf{BSubQ}^*\mathbf{I} \subseteq \mathbf{TxtI}$.

Suppose **M** $\mathbf{BSubQ}^*\mathbf{I}$-identifies $\mathcal{L}$.

Define $\mathbf{M}'(T[m])$ as follows. On input $T[m]$, simulate **M** on input $T[m]$. For each query about language W_i asked at input $T[t]$, answer as follows:

If $(W_{i,m}-\text{content}(T[m])) \cap \{x \mid x < \max(\text{content}(T[t]))\} \neq \emptyset$, then answer no, and give the least element from this set as a counterexample. Otherwise, return yes as the answer.

$\mathbf{M}'$ then outputs the output of **M** on $T[m]$ from the above simulation. This simulation may not always be correct, however note that $(W_{i,m}-\text{content}(T[m])) \cap \{x \mid x < \max(\text{content}(T[t]))\}$, converg-

es to $(W_i - \text{content}(T)) \cap \{x \mid x < \max(\text{content}(T[t]))\}$, as m goes to infinity. Thus, for any question asked by $\mathbf{M}$, for large enough m, the answer given by $\mathbf{M}'$ in simulation of $\mathbf{M}$ is correct. Here, note that after the first question of $\mathbf{M}$ is answered correctly, second question in the simulation must be the "correct question" as asked by $\mathbf{M}$ on input T, and so on. Hence, the conjectures of $\mathbf{M}'$ on T are same as conjectures of $\mathbf{M}$ on T (for **BSubQ*I**-learnability), except for finitely many exceptions. Part (a) follows.

(b) One can show this using proof similar to part (a). Here we use $(W_{i,m} \Delta \text{content}(T[m])) \cap \{x \mid x < \max(\text{content}(T[t]))\}$, instead of $(W_{i,m} - \text{content}(T[m])) \cap \{x \mid x < \max(\text{content}(T[t]))\}$, when giving the answer to equivalence query for the language W_i. Rest of the proof remains essentially the same.

(c) One can show this using proof similar to part (a). We use $(\text{content}(T[m]) - W_{i,m}) \cap \{x \mid x < \max(\text{content}(T[t]))\}$, instead of $(W_{i,m} - \text{content}(T[m])) \cap \{x \mid x < \max(\text{content}(T[t]))\}$, when giving the answer to superset query for the language W_i. □

4. Learning with negative counterexamples to conjectures

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

Definition 17 ([19]). Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ **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:
$T'(n) \in S_n$, if $S_n \neq \emptyset$ and $T'(n) = \#$, if $S_n = \emptyset$,
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) $\mathbf{M}$ **NCEx**^{a}*-identifies* a class $\mathcal{L}$ of languages (written: $\mathcal{L} \subseteq \mathbf{NCEx}^a(\mathbf{M})$), iff $\mathbf{M}$ **NCEx**^{a}-identifies each language in the class.

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

For **LNCEx**^{a} criteria of inference, we consider providing the learner with least counterexample rather than arbitrary one. The criteria **LNCEx**^{a} of learning can thus be defined similarly to **NCEx**^{a}, by requiring $T'(n) = \min(S_n)$, if $S_n \neq \emptyset$ and $T'(n) = \#$, if $S_n = \emptyset$ in clause (a) above (instead of $T'(n)$ being arbitrary member of S_n).

Similarly, one can define $\mathbf{ResNCEx}^a$, where the learner is just told that the latest conjecture is or is not a subset of the input language, but is not provided any counterexamples in the case of "no" answer.

For $\mathbf{BNCEx}^a$ criteria of inference, we update the definition of S_n in clause (a) of the definition of $\mathbf{NCEx}^a$-identification as follows: $S_n = \overline{L} \cap W_{\mathbf{M}(T[n],T'[n])} \cap \{x \mid x \leqslant \max(\text{content}(T[n]))\}$.

We can similarly define $\mathbf{NCBc}^a$, $\mathbf{LNCBc}^a$, $\mathbf{Res\,Bc}^a$ and $\mathbf{BNCBc}^a$ criteria of inference. We refer the reader to [19] for more details, discussion and results about the various variations of **NCI**-criteria.

For $n \in N$, one may also consider the model, $\mathbf{NC}^n\mathbf{I}$, where, for learning a language L, the **NCI** learner is provided counterexamples only for its first n conjectures which are not subsets of L. For remaining conjectures, the answer provided is always #. Following is the formal definition.

Definition 18. Suppose $a \in N \cup \{*\}$, and $m \in N$.

(a) $\mathbf{M}$ $\mathbf{NC}^m\mathbf{Ex}^a$-*identifies a language* L (written: $L \in \mathbf{NC}^m\mathbf{Ex}^a(\mathbf{M})$) iff for all texts T for L, and for all T' satisfying the condition:
$T'(n) \in S_n$, if $S_n \neq \emptyset$ and $\text{card}(\text{content}(T'[n])) < m$; $T'(n) = \#$, if $S_n = \emptyset$ or $\text{card}(\text{content}(T'[n])) \geqslant m$,
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) $\mathbf{M}$ $\mathbf{NC}^m\mathbf{Ex}^a$-*identifies* a class $\mathcal{L}$ of languages (written: $\mathcal{L} \subseteq \mathbf{NC}^m\mathbf{Ex}^a(\mathbf{M})$), iff $\mathbf{M}$ $\mathbf{NC}^m\mathbf{Ex}^a$-identifies each language in the class.

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

For $a \in N \cup \{*\}$ and $\mathbf{I} \in \{\mathbf{Ex}^a, \mathbf{Bc}^a\}$, one can similarly define $\mathbf{BNC}^m\mathbf{I}$ and $\mathbf{LNC}^m\mathbf{I}$ and $\mathbf{NC}^m\mathbf{Bc}^a$.

5. Hierarchies based on the number of queries

Our first two results establish general hierarchies of learning capabilities with respect to the number of queries for all three types of queries. The hierarchy for superset queries is slightly weaker and needs a different proof than hierarchy for other two types of queries. Thus, we separate superset query hierarchy proof from the others.

Theorem 19. *Suppose $n \in N$. Then, there exists a class $\mathcal{L}$ such that*

(a) $\mathcal{L} \in \mathbf{ResNC}^{n+1}\mathbf{Ex} \cap \mathbf{ResSubQ}^{n+1}\mathbf{Ex} \cap \mathbf{ResEquQ}^{n+1}\mathbf{Ex}$.
(b) $\mathcal{L} \notin \mathbf{LSubQ}^n\mathbf{Bc}^* \cup \mathbf{LEquQ}^n\mathbf{Bc}^*$.
(c) $\mathcal{L} \notin \mathbf{LNC}^n\mathbf{Bc}^*$.

Proof. Let $A_{j,k} = [(N - \text{CYL}_1) \cup D_k \cup \{\langle 1, \langle j, k\rangle\rangle\}] - D_j$.

Consider the languages satisfying the following properties:

(I) $1 \leqslant \text{card}(L \cap \text{CYL}_1) \leqslant n + 1$.
(II) $\text{CYL}_0 \subseteq L$.
(III) Either $L = \text{CYL}_0 \cup C$, for some finite set C, or $L = A_{j,k}$, where $\langle 1, \langle j, k\rangle\rangle = \max(L \cap \text{CYL}_1)$.

Let $\mathcal{L}$ denote the collection of languages satisfying the above three properties. Intuitively, the languages in the class $\mathcal{L}$ are either (i) CYL_0 plus finitely many elements, or (ii) a finite variant of $N - \mathrm{CYL}_1$ (where the differences are given by using a *code* in CYL_1). This allows for easy learning, as long as one can check for each possible code $\langle j,k\rangle$, whether the input language is $A_{j,k}$ or not. Usage of $\mathrm{CYL}_0 \subseteq L$, is mainly to ensure that the language is infinite (which is needed to obtain counterexamples for $\mathbf{BNC}^{n+1}\mathbf{Ex}$-learnability).

(a) Consider the following learner $\mathbf{M}$. On input σ, first compute $X = \mathrm{content}(\sigma) \cap \mathrm{CYL}_1$. If X is empty, then output a grammar for $\mathrm{CYL}_0 \cup \mathrm{content}(\sigma)$. Otherwise, let $\langle i, \langle j,k\rangle\rangle = \max(X)$. Query about the language $A_{j,k}$ (if not already done). If the answer is yes, then output a grammar for $A_{j,k}$. Otherwise, output a grammar for $\mathrm{CYL}_0 \cup \mathrm{content}(\sigma)$.

We claim that above $\mathbf{MResSubQ}^{n+1}\mathbf{Ex}$ and $\mathbf{ResEquQ}^{n+1}\mathbf{Ex}$-identifies $\mathcal{L}$. To see this, for any $L \in \mathcal{L}$, note that the algorithm asks at most $n+1$ queries (one for each element in $L \cap \mathrm{CYL}_1$, if and when it is the maximum element in the input data). Furthermore, after the final query (i.e., after $\max(L \cap \mathrm{CYL}_1) = \langle 1, \langle j,k\rangle\rangle$ has been received), based on whether $L = A_{j,k}$ or not (which would have the same answer as whether $A_{j,k} \subseteq L$ or not), the algorithm correctly identifies the input language.

For $\mathbf{ResNC}^{n+1}\mathbf{Ex}$-identification we can use the same method as above, except that this time we conjecture the language $A_{j,k}$ instead of asking a query about this language. If the input language is not $A_{j,k}$, then one would eventually receive a counterexample (note that each language in $\mathcal{L}$ is infinite). Rest of the argument is same as in $\mathbf{ResSubQ}^{1}\mathbf{Ex}$-identification above.

(b) Suppose by way of contradiction that $\mathbf{M}$ witnesses that $\mathcal{L} \in \mathbf{LSubQ}^{n}\mathbf{Bc}^{*}$ ($\mathcal{L} \in \mathbf{LEquQ}^{n}\mathbf{Bc}^{*}$). We show a stronger result: We allow the machine to ask either subset or equivalence queries during its computation, as long as total number of queries is not more than n. Intuitively, in the construction below, we start with one possible *code* in CYL_1 for the diagonalizing language. With each query, we update the code, freezing some of the elements to be in/out of the diagonalizing language. After all queries (which are $\leqslant n$) have been made, we would still have the flexibility that the diagonalizing language could be $\mathrm{CYL}_0 \cup C$, for any finite C (except for the frozen elements) or N–CYL_1 (except for the frozen elements). This would allow for diagonalization using Proposition 11.

Intuitively, j_i, k_i denote the current intended values of j,k as defined in the property (III) for $L \in \mathcal{L}$. Without loss of generality, assume that $D_0 = \emptyset$. Initially let $j_0 = k_0 = 0$, and σ_0 contain $\langle 1, \langle j_0, k_0\rangle\rangle$ as its only element. In the construction we will always have the case that $D_{j_i} \cap (\mathrm{CYL}_0 \cup D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}) = \emptyset$. Intuitively, D_{j_i} denotes the committed negative data, and $D_{k_i}, \langle 1, \langle j_i, k_i\rangle\rangle, \mathrm{CYL}_0$ denote the committed positive data.

Inductively define σ_{i+1} (along with j_{i+1}, k_{i+1}), for $i < n$ as follows.
(* The construction is non-effective. *)
(* The following invariants will be satisfied:
(a) $\mathrm{content}(\sigma_i) \subseteq (\mathrm{CYL}_0 \cup D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\})$.
(b) $D_{j_i} \cap (\mathrm{CYL}_0 \cup D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}) = \emptyset$.
(c) $\mathbf{M}$ has already asked i questions on proper prefixes of σ_i.
(d) Answers given to queries of $\mathbf{M}$ are consistent with any input language L satisfying: $(\mathrm{CYL}_0 \cup D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}) \subseteq L \subseteq N - D_{j_i}$.*)

1. Check if there exists an extension $\sigma \supseteq \sigma_i$, such that $\mathrm{content}(\sigma) \subseteq A_{j_i,k_i}$, and $\mathbf{M}$ asks a question on σ.

If there is no such σ, then $\sigma_{i'}$, $i' > i$ do not get defined.
If there exists such a σ, then choose a shortest such σ, and proceed as follows.

2. Note that A_{j_i,k_i} (and thus σ) does not contain any element of D_{j_i} and CYL_1, except for elements in $D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}$.
Let Q be the queried language.
Let $\sigma_{i+1} = \sigma\#$.
Define j_{i+1}, k_{i+1} and answer the query (with counterexample) based on following cases.
(* We will make sure that $\langle 1, \langle j_{i+1}, k_{i+1}\rangle\rangle > \langle 1, \langle j_i, k_i\rangle\rangle$. *)

3.1. Query is a subset query, and $Q \subseteq \mathrm{CYL}_0 \cup \mathrm{content}(\sigma) \cup D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}$.
In this case give yes answer to the query.
Let $j_{i+1} = j_i$.
Let k_{i+1} be such that $\langle 1, \langle j_{i+1}, k_{i+1}\rangle\rangle > \langle 1, \langle j_i, k_i\rangle\rangle$, and
$D_{k_i} \cup \mathrm{content}(\sigma) \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\} \subseteq D_{k_{i+1}} \subseteq \mathrm{CYL}_0 \cup D_{k_i} \cup \mathrm{content}(\sigma) \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}$.
(* Note that $D_{k_{i+1}}$ only uses committed positive data. *)

3.2. Query is a subset query, and the queried language contains an element not in $\mathrm{CYL}_0 \cup \mathrm{content}(\sigma) \cup D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}$.
Let $w = \min(Q - (\mathrm{CYL}_0 \cup \mathrm{content}(\sigma) \cup D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}))$.
Answer the query as no, and give w as negative data.
Let j_{i+1} be such that $D_{j_{i+1}} = D_{j_i} \cup \{w\}$.
(* For defining k_{i+1}, we need to make sure that w would not interfere with the coding (present or future) in CYL_1. *)
Let k_{i+1} be such that $\langle 1, \langle j_{i+1}, k_{i+1}\rangle\rangle > \max(\{w, \langle 1, \langle j_i, k_i\rangle\rangle\})$, and
$D_{k_i} \cup \mathrm{content}(\sigma) \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\} \subseteq D_{k_{i+1}} \subseteq \mathrm{CYL}_0 \cup D_{k_i} \cup \mathrm{content}(\sigma) \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}$.

3.3. Query is an equivalence query.
Let w be the least number such that one of the following properties is satisfied.
(A) $w \in N - (\mathrm{CYL}_0 \cup \mathrm{CYL}_1 \cup D_{j_i} \cup D_{k_i} \cup \mathrm{content}(\sigma))$.
(* That is w is outside the committed or coding area. *)
(B) $w \in (\mathrm{CYL}_0 \cup \mathrm{content}(\sigma) \cup D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}) - Q$.
(C) $w \in Q \cap [(\mathrm{CYL}_1 \cup D_{j_i}) - (D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\})]$.
Answer the query as no, and give w as the counterexample.
If $w \in Q$, then
Let j_{i+1} be such that $D_{j_{i+1}} = D_{j_i} \cup \{w\} \cup \{\langle 1, x\rangle < w \mid \langle 1, x\rangle \notin D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}\}$.
(* We need to add $\{\langle 1, x\rangle < w \mid \langle 1, x\rangle \notin D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}\}$ so that the counterexample w above is indeed the least counterexample, for any possible L as in invariant (d) above. *)
Let k_{i+1} be such that $\langle 1, \langle j_{i+1}, k_{i+1}\rangle\rangle > \max(\{w, \langle 1, \langle j_i, k_i\rangle\rangle\})$, and
$D_{k_i} \cup \mathrm{content}(\sigma) \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\} \subseteq D_{k_{i+1}} \subseteq \mathrm{CYL}_0 \cup D_{k_i} \cup \mathrm{content}(\sigma) \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}$.
Else (i.e., $w \notin Q$),
Let j_{i+1} be such that $D_{j_{i+1}} = D_{j_i} \cup \{\langle 1, x\rangle < w \mid \langle 1, x\rangle \notin D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}\}$.
Let k_{i+1} be such that $\langle 1, \langle j_{i+1}, k_{i+1}\rangle\rangle > \max(\{w, \langle 1, \langle j_i, k_i\rangle\rangle\})$, and
$D_{k_i} \cup \{w\} \cup \mathrm{content}(\sigma) \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\} \subseteq D_{k_{i+1}} \subseteq \mathrm{CYL}_0 \cup D_{k_i} \cup \{w\} \cup \mathrm{content}(\sigma) \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\}$.

End

It is easy to verify that the above construction maintains the invariants.

Now let m be the largest number such that σ_m is defined. Note that $\mathbf{M}$ does not make any further queries on any $\sigma \supseteq \sigma_m$, such that content$(\sigma) \subseteq A_{j_m,k_m}$ (if $m = n$, due to bound on number of queries, $\mathbf{M}$ cannot make any more queries; if $m < n$, the failure of search for $\sigma \supseteq \sigma_m$, on which $\mathbf{M}$ asks a query, implies that $\mathbf{M}$ does not make any more queries). Thus, $\mathbf{M}$ needs to $\mathbf{Bc}^*$-identify A_{j_m,k_m} and $L = \mathrm{CYL}_0 \cup D_{k_m} \cup \{\langle 1, \langle j_m, k_m\rangle\rangle\} \cup C$, for all finite C such that $C \subseteq A_{j_m,k_m}$. This is impossible by Proposition 11.

(c) This can be done in a way similar to part (b), except that we do not consider queries, but consider conjectures by the learner. We search for σ such that the conjectured language contains an element not in content$(\sigma) \cup D_{k_i} \cup \{\langle 1, \langle j_i, k_i\rangle\rangle\} \cup \mathrm{CYL}_0$, and when such σ is found, we define σ_{i+1}, j_{i+1}, k_{i+1}, and the counterexample as in step 3.2 above. We omit the details.

Theorem follows from the above analysis. □

$\mathcal{L}$ used in Theorem 19 can also be shown to be in $\mathbf{BNC}^{n+1}\mathbf{EX} - \mathbf{BNC}^{n}\mathbf{Ex}$.

We now turn our attention to the hierarchy based on the number of superset queries. As $\mathbf{LSupQ}^*\mathbf{Bc}^* \subseteq \mathbf{TxtBc}^*$ (see Theorem 57), the hierarchy for superset queries takes a slightly weaker form than hierarchies for other types of queries.

The following lemma is useful in proving Theorem 22, as well as some other theorems involving superset queries below.

Lemma 20. *There exists a recursive F (which takes as input a number e, a finite set S, a machine $\mathbf{M}$) such that one of the following is satisfied*:

(a) $W_{F(e,S,\mathbf{M})}$ is infinite and $S \cup \{\langle e,x\rangle \mid x \in W_{F(e,S,\mathbf{M})}\} \notin \bigcup_{t\in N} \mathbf{TxtBc}^t(\mathbf{M})$, or

(b) $W_{F(e,S,\mathbf{M})}$ is finite, and for some $w \in N$, for some $S' \subseteq \{\langle e,x\rangle \mid x < 2w\}$ such that $(\forall x < w)[S' \cap \{\langle e,2x\rangle, \langle e,2x+1\rangle\} \neq \emptyset]$, $S \cap \{\langle e,2w\rangle, \langle e,2w+1\rangle\} = \emptyset$, and

$$S \cup S' \cup \{\langle e,2y\rangle \mid y > w\} \notin \bigcup_{t\in N} \mathbf{TxtBc}^t(\mathbf{M})$$

Proof. $W_{F(e,S,\mathbf{M})}$ is defined as follows. Initially, let σ_0 be such that content$(\sigma_0) = S$. Let $B_0 = \emptyset$. Intuitively, B_s denotes the set of elements which we have decided to keep out of $W_{F(e,S,\mathbf{M})}$. Let $W^s_{F(e,S,\mathbf{M})}$ denote $W_{F(e,S,\mathbf{M})}$ enumerated before stage s.

We will maintain the following invariants:

(i) For any x, B_s contains at most one of $\{2x, 2x+1\}$.
(ii) content(σ_s) is $S \cup \{\langle e,x\rangle \mid x \in W^s_{F(e,S,\mathbf{M})}\}$.
(iii) $(S \cup \{\langle e,x\rangle \mid x \in W_{F(e,S,\mathbf{M})}\}) \cap \{\langle e,x\rangle \mid x \in B_s\} = \emptyset$.

Go to stage 0.

Stage s

1. Search for a $\sigma \supseteq \sigma_s$, such that content$(\sigma) \subseteq S \cup \{\langle e,x\rangle \mid x \notin B_s\}$ and there exists a set A of cardinality $s+1$ with the following properties:

(a) $\{\langle e,x\rangle \mid x \in A\} \subseteq W_{\mathbf{M}(\sigma)}$.
(b) $\text{content}(\sigma) \cap \{\langle e,x\rangle \mid x \in A\} = \emptyset$.
(c) For all x, $A \cup B_s$ contains at most one element from $\{2x, 2x+1\}$.

2. If and when such σ and A are found, let
$B_{s+1} = B_s \cup A$.
$W^{s+1}_{F(e,S,\mathbf{M})} = W^{s}_{F(e,S,\mathbf{M})} \cup \{x \mid \langle e,x\rangle \in \text{content}(\sigma)\} \cup \{w\}$, where w is the least element such that $w \notin B_s \cup A$ and $w > s$.
Let σ_{s+1} be an extension of σ such that $\text{content}(\sigma_{s+1}) = \text{content}(\sigma) \cup \{\langle e,w\rangle\}$.
Go to stage $s+1$.

End stage s

It is easy to verify that invariants are satisfied. Now consider the following cases.

Case 1: There are infinitely many stages.

In this case let $T = \bigcup_{s\in N} \sigma_s$. Let $B = \bigcup_{s\in N} B_s$. It is easy to see that $W_{F(e,S,\mathbf{M})}$ is infinite (due to addition of arbitrarily large w to $W_{F(e,S,\mathbf{M})}$ in step 2 for each stage).

Furthermore, for every t, $\mathbf{M}$ on T outputs infinitely many conjectures (at σ found at each stage $s > t$) which enumerate at least $t+1$ elements from $\{\langle e,x\rangle \mid x \in B\}$. Thus, $\mathbf{M}$ does not $\mathbf{TxtBc}^t$-identify $S \cup \{\langle e,x\rangle \mid x \in W_{F(e,S,\mathbf{M})}\}$. (Note that $\{\langle e,x\rangle \mid x \in B\}$ does not intersect with $S \cup \{\langle e,x\rangle \mid x \in W_{F(e,S,\mathbf{M})}\}$, due to invariant (iii) mentioned above.) Thus, clause (a) in the lemma holds.

Case 2: Stage s starts but does not end.

In this case let w be such that $w > \max(\{w' \mid \langle e,w'\rangle \in S \text{ or } w' \in W^{s}_{F(e,S,\mathbf{M})} \cup B_s\})$. Now consider the language

$L = S \cup \{\langle e,x\rangle \mid x < 2w, x \notin B_s\} \cup \{\langle e,2x\rangle \mid x > w\}$.

Now, $\mathbf{M}$ on any $\sigma \supseteq \sigma_s$, such that $\text{content}(\sigma) \subseteq L$, outputs at most finitely many elements from L (otherwise search in step 1 would have succeeded). Thus, for all t, $\mathbf{M}$ does not $\mathbf{TxtBc}^t$-identify L. Thus, clause (b) in the lemma holds.

From the above cases lemma follows. □

Corollary 21. *There exists a recursive F (which takes as input a number e, a finite set S, a machine $\mathbf{M}$) such that one of the following is satisfied*:

(a) $W_{F(e,S,\mathbf{M})}$ is infinite and $S \cup \{\langle e,x\rangle \mid x \in W_{F(e,S,\mathbf{M})}\} \notin \mathbf{TxtEx}^*(\mathbf{M})$, or

(b) $W_{F(e,S,\mathbf{M})}$ is finite, and for some $w \in N$, for some $S' \subseteq \{\langle e,x\rangle \mid x < 2w\}$ such that $(\forall x < w)[S' \cap \{\langle e,2x\rangle, \langle e,2x+1\rangle\} \neq \emptyset]$, $S \cap \{\langle e,2w\rangle, \langle e,2w+1\rangle\} = \emptyset$, and

$$S \cup S' \cup \{\langle e,2y\rangle \mid y > w\} \notin \mathbf{TxtEx}^*(\mathbf{M}).$$

Now we exhibit the hierarchy for superset queries.

Theorem 22. *For all $n \in N$, there exists a $\mathcal{L}$ such that*

(a) for all $t \in N$, $\mathcal{L} \notin \mathbf{LSupQ}^n\mathbf{Bc}^t$;
(b) $\mathcal{L} \notin \mathbf{LSupQ}^n\mathbf{Ex}^*$;
(c) $\mathcal{L} \in \mathbf{ResSupQ}^{n+1}\mathbf{Ex}$.

Proof. Consider the following class of languages.

$\mathcal{L} = \{L \mid (\exists r \leqslant n)[$
  Let $S = \{i \mid L \cap \mathrm{CYL}_i \neq \emptyset\}$.
  Let $e = \max(S)$.
1. $\mathrm{card}(S) = 2r + 1$.
2. $(L - \mathrm{CYL}_e)$ is finite.
3. Either
   3.1 W_e is infinite and $L \cap \mathrm{CYL}_e = \{\langle e, x\rangle \mid x \in W_e\}$.
   or
   3.2 W_e is finite, and $(\exists w)[$
   $L \cap \{\langle e, 2w\rangle, \langle e, 2w+1\rangle\} = \emptyset$ and
   $(\forall x < w)[L \cap \{\langle e, 2x\rangle, \langle e, 2x+1\rangle\} \neq \emptyset]$, and
   $(\forall y > w)[\langle e, 2y\rangle \in L \ \wedge \ \langle e, 2y+1\rangle \notin L]]$.
   ]}

Claim 23. $\mathcal{L} \in \mathbf{ResSupQ}^{n+1}\mathbf{Ex}$.

Proof. We first describe the queries made by the learner.

On input σ, the learner first calculates $S = \{j \mid \mathrm{content}(\sigma) \cap \mathrm{CYL}_j \neq \emptyset\}$. Let $e = \max(S)$. If $\mathrm{card}(S) = 2r + 1$, for some $r \leqslant n$, then make the query (if not already made) about whether:

$$(N - \mathrm{CYL}_e) \cup \{\langle e, x\rangle \mid x \in W_e\}$$

is a superset of the input language.

Note that above process would make at most $n + 1$ queries on texts for languages from $\mathcal{L}$, one for each possible $r \leqslant n$. Now suppose T is a text for $L \in \mathcal{L}$. A learner can make the queries as above, and thus in the limit will

(i) compute $S = \{j \mid L \cap \mathrm{CYL}_j \neq \emptyset\}$,
(ii) compute $e = \max(S)$,
(iii) know whether

$$(N - \mathrm{CYL}_e) \cup \{\langle e, x\rangle \mid x \in W_e\}$$

is a superset of L. Now consider the following cases:

Case 1: $(N - \mathrm{CYL}_e) \cup \{\langle e, x\rangle \mid x \in W_e\}$ is superset of L.

The learner outputs (in the limit on T) a grammar for $[\mathrm{content}(T) \cap (N - \mathrm{CYL}_e)] \cup \{\langle e, x\rangle \mid x \in W_e\}$.

Case 2: $(N - \mathrm{CYL}_e) \cup \{\langle e, x\rangle \mid x \in W_e\}$ is not a superset of L.

The learner computes, in the limit, the least w such that both $\langle e, 2w\rangle$ and $\langle e, 2w+1\rangle$ do not belong to L (if $L \in \mathcal{L}$, then there must exist such a w).

The learner outputs, in the limit on T, a grammar for $[\mathrm{content}(T) \cap (N - \mathrm{CYL}_e)] \cup \{\langle e, x\rangle \mid x < 2w, \langle e, x\rangle \in \mathrm{content}(T)\} \cup \{\langle e, 2x\rangle \mid x > w\}$.

It is easy to verify that the above learner would $\mathbf{ResSupQ}^{n+1}\mathbf{Ex}$ -identify $\mathcal{L}$.

Claim 24. *(a) For all* $t \in N$, $\mathcal{L} \notin \mathbf{LSupQ}^n\mathbf{Bc}^t$.
(b) $\mathcal{L} \notin \mathbf{LSupQ}^n\mathbf{Ex}^*$.

Proof. We only show part (a). Part (b) can be shown using Corollary 21 instead of using Lemma 20.

Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{LSupQ}^n\mathbf{Bc}^t$-identifies $\mathcal{L}$. We first define σ_i, and finite sets S_i as follows. Initially, $S_i = \emptyset$ and $\sigma_0 = \Lambda$.

Inductively define σ_{i+1}, S_{i+1}, for $i < n$ as follows.
(* The construction is non-effective. *)
(* We will have the following invariants:
(a) $\mathrm{card}(S_i) = 2i$.
(b) $S_i = \{ j \mid \mathrm{content}(\sigma_i) \cap \mathrm{CYL}_j \neq \emptyset \}$.
(c) $\mathbf{M}$ has already asked at least i queries on proper prefixes of σ_i.
(d) Answers given to $\mathbf{M}$ on queries made on proper prefixes of σ_i are consistent with any language L such that $\mathrm{content}(\sigma_i) \subseteq L$.*)

1. Check if there exists a $\sigma \supseteq \sigma_i$ such that, for some $e \notin S_i$, $\mathrm{content}(\sigma) \subseteq \bigcup_{j \in S_i \cup \{e\}} \mathrm{CYL}_e$ and $\mathbf{M}$ asks a query on σ.
   If there is no such σ, then $\sigma_{i'}, S_{i'}$ for $i' > i$ do not get defined.
   If there exists such a σ, then fix a shortest such σ and corresponding e and proceed as follows.
2. Suppose the queried language is Q.
3. If $Q = N$, then answer the query as yes.
   Let j be arbitrary element not in $S_i \cup \{e\}$.
   Let $S_{i+1} = S_i \cup \{e, j\}$.
   Let σ_{i+1} be an extension of σ such that $\mathrm{content}(\sigma_{i+1}) = \mathrm{content}(\sigma) \cup \{\langle e, 0\rangle, \langle j, 0\rangle\}$.
   (* We added $\langle e, 0\rangle$ just to make sure that σ_{i+1} contains at least one element from CYL_e. $\langle j, 0\rangle$ is added to make σ_{i+1} contain elements from $2(i+1)$ cylinders, for satisfying the invariant (b). *)
4. If $Q \neq N$, then answer the query as no, with $\langle r, r'\rangle = \min(N - Q)$ as the counterexample.
   If $r \notin S_i \cup \{e\}$, then let $j = r$. Otherwise, let j be arbitrary element not in $S_i \cup \{e\}$.
   Let $S_{i+1} = S_i \cup \{e, j\}$.
   Let σ_{i+1} be an extension of σ such that $\mathrm{content}(\sigma_{i+1}) = \mathrm{content}(\sigma) \cup \{\langle e, 0\rangle, \langle j, 0\rangle, \langle r, r'\rangle\}$.
   (* We added $\langle e, 0\rangle$ just to make sure that σ_{i+1} contains at least one element from CYL_e. $\langle j, 0\rangle$ is added to make σ_{i+1} contain elements from $2(i+1)$ cylinders, for satisfying the invariant (b). *)
   (* We assume without loss of generality that if $\sigma \subset \sigma' \subseteq \sigma_{i+1}$, then $\mathbf{M}$ does not ask any questions. If not, then one can just delay these questions beyond σ_{i+1}, without effecting this construction. *)

End

It is easy to verify that invariants are maintained by the construction. Let m be maximal such that σ_m is defined. Now $\mathbf{M}$ on any extension σ of σ_m, such that $\sigma \subseteq \bigcup_{j \in S_m \cup \{e\}} \mathrm{CYL}_j$, for some e, does not ask any more questions. Thus, one can design a machine $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{TxtBc}^t$-identifies all L such that $\mathbf{M}$ $\mathbf{LSupQ}^n\mathbf{Bc}^t$ identifies L and $\mathrm{content}(\sigma_m) \subseteq L \subseteq \bigcup_{j \in S_m \cup \{e\}} \mathrm{CYL}_j$, for some e.

Now, let F be as in Lemma 20. By Kleene's recursion theorem [28], there exists an $e > \max(S_m)$, such that $W_e = W_{F(e, \mathrm{content}(\sigma_m), \mathbf{M}')}$. It now follows from Lemma 20 that $\mathbf{M}'$ does not $\mathbf{TxtBc}^t$-identify some language $L \in \mathcal{L}$, such that $\mathrm{content}(\sigma_m) \subseteq L$. Thus, $\mathbf{M}$ does not $\mathbf{LSupQ}^n\mathbf{Bc}^t$-identify L and hence $\mathcal{L}$. □

6. Hierarchies based on type of counterexamples

6.1. One query: least counterexamples do no better than no counterexamples

Before turning our attention to hierarchies based on the type of counterexamples, we first show that, when only a single query is used, different types of counterexamples do not make a difference.

Theorem 25. *Suppose* $a \in N \cup \{*\}$, *and* $\mathbf{I} \in \{\mathbf{Ex}^a, \mathbf{Bc}^a\}$.
(a) $\mathbf{ResSubQ}^1\mathbf{I} = \mathbf{SubQ}^1\mathbf{I} = \mathbf{LSubQ}^1\mathbf{I}$.
(b) $\mathbf{ResNC}^1\mathbf{I} = \mathbf{NC}^1\mathbf{I} = \mathbf{LNC}^1\mathbf{I}$.
(c) $\mathbf{ResEquQ}^1\mathbf{I} = \mathbf{EquQ}^1\mathbf{I} = \mathbf{LEquQ}^1\mathbf{I}$.
(d) $\mathbf{ResSupQ}^1\mathbf{I} = \mathbf{SupQ}^1\mathbf{I} = \mathbf{LSupQ}^1\mathbf{I}$.

Proof. (a) Since $\mathbf{Res\ SubQ}^1\mathbf{I} \subseteq \mathbf{SubQ}^1\mathbf{I} \subseteq \mathbf{LSubQ}^1\mathbf{I}$, it suffices to show that $\mathbf{LSubQ}^1\mathbf{I} \subseteq \mathbf{ResSubQ}^1\mathbf{I}$.

Suppose $\mathbf{M}$ $\mathbf{LSubQ}^1\mathbf{I}$ -identifies $\mathcal{L}$. We assume without loss of generality that $\mathbf{M}$ never asks more than 1 query whatever the input or answers (even if the answer is wrong, or language is outside the class being learned).

Define $\mathbf{M}'$ as follows. On input $T[n]$, simulate $\mathbf{M}(T[n])$. For the only query, if any, about a language W_i answer as follows. If the answer received by $\mathbf{M}'$ for the same query is yes, then return yes as the answer. If the answer received by $\mathbf{M}'$ is no, then answer no, along with $\min(W_{i,n} - \text{content}(T[n]))$ as the counterexample.

$\mathbf{M}'$ then outputs the output of $\mathbf{M}$ on $T[n]$, using the above simulation. This simulation may not always be correct, however note that if $W_i - \text{content}(T) \neq \emptyset$, then $\min(W_{i,n} - \text{content}(T[n]))$ converges to $\min(W_i - \text{content}(T))$, as n goes to infinity. Thus, for large enough n, the answer given by $\mathbf{M}'$ in simulation of $\mathbf{M}$ is correct. Hence, the sequence of conjectures of $\mathbf{M}'$ on T are same as the sequence of conjectures of $\mathbf{M}$ on T (for $\mathbf{LSubQ}^1\mathbf{I}$ - learnability), except for finitely many exceptions. Part (a) follows.

Part (b) can be proved in a way similar to (a).

(c) One can show this using proof similar to part (a). We use $\min(W_{i,n} \Delta \text{content}(T[n]))$ instead of $\min(W_{i,n} - \text{content}(T[n]))$ when giving the answer to equivalence query for the language W_i. Rest of the proof remains essentially the same.

(d) One can show this using proof similar to part (a). We use $\min(\text{content}(T[n]) - W_{i,n})$ instead of $\min(W_{i,n} - \text{content}(T[n]))$ when giving the answer to superset query for the language W_i. Rest of the proof remains essentially the same. □

The above theorem thus restricts us to consider at least two queries when showing differences between various types of counterexamples. The next two sections will address these differences.

6.2. Advantages of having least counterexamples

We first consider equivalence and subset queries. Our result shows that **Ex**-learners using just two subset or equivalence queries and receiving the least counterexamples can sometimes do better than any $\mathbf{Bc}^*$-learner making any n queries of either type and receiving arbitrary counterexamples.

Theorem 26. *For all* $n \in N$, $\mathbf{LSubQ}^2\mathbf{Ex} \cap \mathbf{LEquQ}^2\mathbf{Ex} - (\mathbf{SubQ}^n\mathbf{Bc}^* \cup \mathbf{EquQ}^n\mathbf{Bc}^*) \neq \emptyset$.

Proof. Define $\mathcal{L}$ as follows.

$\mathcal{L} = \{L \mid (\exists m > 0)[$
1. $\{\langle 0, x\rangle \mid x < m\} = L \cap \mathrm{CYL}_0$, and
2. $L \cap \{y \mid y \leqslant \langle 0, m\rangle\} = \{\langle 0, x\rangle \mid x < m\}$, and
3. $\mathrm{card}(L \cap \mathrm{CYL}_1) = m$, and
4. Suppose $A = \{j \mid (\exists k)[\langle 1, \langle j, k\rangle\rangle \in L]\}$. Then $\min(A) > 1$. Furthermore,
   - 4.1 For $j \in A$, $j \neq \max(A)$, $\mathrm{CYL}_j \subseteq L$.
   - 4.2 Either $\mathrm{CYL}_{\max(A)} \subseteq L$ or L contains only finitely many elements from $\mathrm{CYL}_{\max(A)}$.
   - 4.3. If $j \notin A \cup \{0, 1\}$, then L does not contain any elements from CYL_j.

]}

Intuitively, for $L \in \mathcal{L}$, CYL_0 portion of the language (i.e, the part $\mathrm{CYL}_0 \cap L$) codes a value m. Then there are exactly m different elements in CYL_1, indicating which cylinders are present in L. All except possibly one of these cylinders is present fully in L. The remaining one is used to achieve the diagonalization.

Claim 27. $\mathcal{L} \in \mathbf{LSubQ}^2\mathbf{Ex} \cap \mathbf{LEquQ}^2\mathbf{Ex}$.

Proof. A learner initially asks a query about whether the input language contains (is equivalent to) CYL_0. Since CYL_0 is not a subset of any language in the class, learner will receive a least counterexample (for both learning via subset queries or learning via equivalence queries). Note that due to clause 2 in the definition of $\mathcal{L}$, this least counterexample must be from CYL_0. Suppose the counterexample received is $\langle 0, m\rangle$. Then, the learner waits until it has received exactly m distinct elements of CYL_1 in the input. Then, the learner computes, $X = L \cap \mathrm{CYL}_1$ and $A = \{j \mid (\exists k)[\langle 1, \langle j, k\rangle\rangle \in X]\}$ (note that after m elements have already been received, for language L in the class, A can be computed). Then, $\mathbf{M}$ asks a query about the language $\{\langle 0, x\rangle \mid x < m\} \cup X \cup \bigcup_{j \in A} \mathrm{CYL}_j$. If the answer is yes (either for subset or for equivalence query), then the input language must be $\{\langle 0, x\rangle \mid x < m\} \cup X \cup \bigcup_{j \in A} \mathrm{CYL}_j$. On the other hand, if the answer is no, then the input language must be of form $\{\langle 0, x\rangle \mid x < m\} \cup X \cup C \cup \bigcup_{j \in A, j \neq \max(A)} \mathrm{CYL}_j$, for some finite set $C \subseteq \mathrm{CYL}_{\max(A)}$. One can determine this C from the input in the limit, without asking any more questions. Thus, $\mathcal{L} \in \mathbf{LSubQ}^2\mathbf{Ex} \cap \mathbf{LEquQ}^2\mathbf{Ex}$.

Claim 28. $\mathcal{L} \notin \mathbf{EquQ}^n\mathbf{Bc}^* \cup \mathbf{SubQ}^n\mathbf{Bc}^*$.

Proof. We will show a stronger claim. We let the machine $\mathbf{M}$ ask queries of either subset or equivalence type. However the total number of queries must be limited to n. So suppose by way of contradiction that $\mathbf{M}$ $\mathbf{Bc}^*$-identifies $\mathcal{L}$ using n queries.

We will maintain two variables, l_i and u_i, which will indicate that any value of m (as in the definition of $\mathcal{L}$) satisfying $l_i \leqslant m \leqslant u_i$ would be consistent with the data σ_i and the answers given to queries upto now. We will also maintain sets A_i, X_i (intuitively, $X_i \subseteq \mathrm{CYL}_1$ would be committed to belong to L, and A_i would represent the set we intend to use for A, as in the definition of $\mathcal{L}$, for the diagonalizing language L).

Initially, let $\sigma_0 = \Lambda$. Let $l_0 = 1, u_0 = 2^{n+2} - 1$. Let $A_0 = \{j_0\}$, $X_0 = \{\langle 1, \langle j_0, k_0\rangle\rangle\}$, where j_0, k_0 are large enough so that $j_0 > 1$, as well as $\langle 1, \langle j_0, k_0\rangle\rangle$ and $\langle j_0, 0\rangle$ are both $> \langle 0, u_0\rangle$.

Inductively define $\sigma_{i+1}, l_{i+1}, u_{i+1}, A_{i+1}, X_{i+1}$, for $i < n$ as follows.
(* The construction is non-effective. *)
(* Following invariants will be satisfied:
(a) $u_i - l_i = 2^{n+2-i} - 2$.
(b) $A_i \cap \{0,1\} = \emptyset$ and $X_i \subseteq \mathrm{CYL}_1$. Moreover, for any element $\langle 1, \langle j,k\rangle\rangle \in X_i$, we have $\langle j, 0\rangle$ and $\langle 1, \langle j,k\rangle\rangle$ are both greater than $\langle 0, u_i\rangle$
(c) $\mathrm{card}(X_i) = l_i$.
(d) $A_i = \{j \mid (\exists k)[\langle 1, \langle j,k\rangle\rangle \in X_i]\}$.
(e) $\mathrm{content}(\sigma_i) \subseteq \{\langle 0,x\rangle \mid x < l_i\} \cup X_i \cup \bigcup_{r \in A_i - \{\max(A_i)\}} \mathrm{CYL}_r$.
(f) **M** has already asked i queries on proper prefixes of σ_i.
(g) Answers given to **M** are consistent with any input language L which satisfies:

$$\{\langle 0,x\rangle \mid x < l_i\} \cup X_i \cup \bigcup_{r \in A_i - \{\max(A_i)\}} \mathrm{CYL}_r$$

$$\subseteq L \subseteq$$

$$\{\langle 0,x\rangle \mid x < u_i\} \cup X_i \cup \{\langle 1, \langle j,k\rangle\rangle \mid \langle 1, \langle j,k\rangle\rangle > \max(X_i), j > \max(A_i)\} \cup$$

$$\bigcup_{r \in A_i \text{ or } r > \max(A_i)} \mathrm{CYL}_r.$$

*)

1. Check if there exists a σ extending σ_i such that $\mathrm{content}(\sigma) \subseteq \{\langle 0,x\rangle \mid x < l_i\} \cup X_i \cup \bigcup_{j \in A_i} \mathrm{CYL}_j$ and **M** asks a query on σ.
   If there is no such σ, then $\sigma_{i'}$, $i' > i$ do not get defined.
   If there exists such a σ, then choose a shortest such σ, and proceed as follows.
2. Let Q be the queried language. Let $\sigma_{i+1} = \sigma\#$.
   Define $l_{i+1}, u_{i+1}, A_{i+1}, X_{i+1}$ based on the following cases.

2.1 **M**(σ) asked an equivalence query on σ.
   In this case, if Q contains $\langle 0, \frac{l_i+u_i}{2}\rangle$,
   Then let $l_{i+1} = l_i$ and $u_{i+1} = \frac{l_i+u_i}{2} - 1$.
   Else let $l_{i+1} = \frac{l_i+u_i}{2} + 1$, $u_{i+1} = u_i$.
   Give answer no to the query, and give $\langle 0, \frac{l_i+u_i}{2}\rangle$ as a counterexample.
   (* Note that, in the If case the counterexample was negative, whereas in the Else case, the counterexample was positive. *)
   Let $S \subseteq \mathrm{CYL}_1$ be such that $\mathrm{card}(S) = l_{i+1} - l_i$, and for all $\langle 1, \langle j,k\rangle\rangle \in S$, $j > \max(A_i)$ and $\langle 1, \langle j,k\rangle\rangle > \max(X_i)$.
   (* Note that, if $l_{i+1} = l_i$, then S is empty. *)
   Let $X_{i+1} = X_i \cup S$.
   Let $A_{i+1} = \{j \mid (\exists k)[\langle 1, \langle j,k\rangle\rangle \in X_{i+1}]\}$.
   (* Note that adding S as above to the diagonalizing language makes sure that, $\mathrm{card}(X_{i+1}) = l_{i+1}$ as required in the invariant (c). *)

2.2 $\mathbf{M}(\sigma)$ asks a subset query on input σ and $Q - (\text{content}(\sigma) \cup X_i \cup \{\langle 0,x\rangle \mid x < u_i\} \cup \bigcup_{j\in A_i} \text{CYL}_j) \neq \emptyset$.

Let $\langle w,z\rangle$ be an element of $Q - (\text{content}(\sigma) \cup X_i \cup \{\langle 0,x\rangle \mid x < u_i\} \cup \bigcup_{j\in A_i} \text{CYL}_j)$.

Give the answer no and provide $\langle w,z\rangle$ as a counterexample.

(* Note that we update the variables to maintain the invariants mentioned above. In particular for invariant (g), we need to ensure that the elements added to X_{i+1} and A_{i+1} are large enough, compared to the counterexample given above. *)

Let $l_{i+1} = \frac{l_i+u_i}{2} + 1$, $u_{i+1} = u_i$.

Let $S \subseteq \text{CYL}_1$ be such that

$\text{card}(S) = l_{i+1} - l_i$, and for all $\langle 1, \langle j,k\rangle\rangle \in S$, $j > \max(A_i \cup \{w\})$ and $\langle 1, \langle j,k\rangle\rangle > \max(X_i \cup \{\langle w,z\rangle\})$.

Let $X_{i+1} = X_i \cup S$.

Let $A_{i+1} = \{j \mid (\exists k)[\langle 1, \langle j,k\rangle\rangle \in X_{i+1}]\}$.

2.3 $\mathbf{M}(\sigma)$ asks a subset query for language Q, and $Q \subseteq (\text{content}(\sigma) \cup X_i \cup \{\langle 0,x\rangle \mid x < u_i\} \cup \bigcup_{j\in A_i} \text{CYL}_j)$.

If Q contains an element of form $\langle 0,x\rangle, x > \frac{l_i+u_i}{2}$,

Then give answer no to the query and provide $\langle 0,x\rangle$ as the counterexample.

Let $u_{i+1} = \frac{l_i+u_i}{2} - 1$, $l_{i+1} = l_i$.

If Q does not contain an element of form $\langle 0,x\rangle, x > \frac{l_i+u_i}{2}$,

Then give answer yes to the query.

Let $u_{i+1} = u_i$, $l_{i+1} = \frac{l_i+u_i}{2} + 1$.

Let $S \subseteq \text{CYL}_1$ be such that $\text{card}(S) = l_{i+1} - l_i$, and for all $\langle 1, \langle j,k\rangle\rangle \in S$, $j > \max(A_i)$ and $\langle 1, \langle j,k\rangle\rangle > \max(X_i)$.

Let $X_{i+1} = X_i \cup S$.

Let $A_{i+1} = \{j \mid (\exists k)[\langle 1, \langle j,k\rangle\rangle \in X_{i+1}]\}$.

End

It is easy to verify that invariants are maintained by the above construction. Thus, $u_i > l_i$, for $i \leqslant n$. Now, let m be largest number such that σ_m is defined. Clearly, $\mathbf{M}$ does not ask any further questions on $\sigma \supseteq \sigma_m$, such that $\text{content}(\sigma) \subseteq \{\langle 0,x\rangle \mid x < l_m\} \cup X_m \cup \bigcup_{j\in A_m} \text{CYL}_j$ (either $m = n$, in which case $\mathbf{M}$ has already asked n questions, or the search for σ in the above construction did not succeed for $i = m$). Thus, $\mathbf{M}$ needs to $\mathbf{Bc}^*$-identify, without any further questions, the language $\{\langle 0,x\rangle \mid x < l_m\} \cup X_m \cup \bigcup_{j\in A_m} \text{CYL}_j$, and also the languages $\text{content}(\sigma_m) \cup \{\langle 0,x\rangle \mid x < l_m\} \cup X_m \cup S \cup \bigcup_{j\in A_m, j\neq \max(A_m)} \text{CYL}_j$, for every finite $S \subseteq \text{CYL}_{\max(A_m)}$. This is not possible by Proposition 11. □

The following theorem shows that **Ex**-learners using just two superset queries and getting least counterexamples can sometimes do better than any $\mathbf{Bc}^t$-learner ($t \in N$) using n superset queries and getting arbitrary counterexamples. Note, though, that this theorem cannot be generalized for diagonalization against $\mathbf{SupQ}^n\mathbf{Bc}^*$ (as $\mathbf{LSupQ}^*\mathbf{Bc}^* \subseteq \mathbf{TxtBc}^*$, see Theorem 57) or against $\mathbf{SupQ}^*\mathbf{Ex}$ (as $\mathbf{LSupQ}^*\mathbf{I} = \mathbf{SupQ}^*\mathbf{I} = \mathbf{ResSupQ}^*\mathbf{I}$, see Proposition 41).

Theorem 29. *For all $n \in N$, there exists a $\mathcal{L}$ such that*

(a) for all $t \in N$, $\mathcal{L} \notin \mathbf{SupQ}^n\mathbf{Bc}^t$;

(b) $\mathcal{L} \notin \mathbf{SupQ}^n\mathbf{Ex}^*$;

(c) $\mathcal{L} \in \mathbf{LSupQ}^2\mathbf{Ex}$.

Proof. Consider the following class of languages.

$\mathcal{L} = \{L \mid (\exists r \mid 1 \leqslant r \leqslant 3^{n+2}+1)[$
1. $L \cap \mathrm{CYL}_0 = \{\langle 0, x\rangle \mid 3^{n+2}+2-r \leqslant x \leqslant 3^{n+2}+1\}$.
   Let $S = \{i > 0 \mid L \cap \mathrm{CYL}_i \neq \emptyset\}$.
   Let $e = \max(S)$.
2. $\mathrm{card}(S) = r$.
3. $(L - \mathrm{CYL}_e)$ is finite.
4. Either
   4.1 W_e is infinite and $L \cap \mathrm{CYL}_e = \{\langle e, x\rangle \mid x \in W_e\}$.
   or
   4.2 W_e is finite, and $(\exists w)[$
   $L \cap \{\langle e, 2w\rangle, \langle e, 2w+1\rangle\} = \emptyset$ and
   $(\forall x < w)[L \cap \{\langle e, 2x\rangle, \langle e, 2x+1\rangle\} \neq \emptyset]$, and
   $(\forall y > w)[\langle e, 2y\rangle \in L \ \wedge \ \langle e, 2y+1\rangle \notin L]]$.
   ]}

Intuitively, $L \in \mathcal{L}$ would contain elements from r of the cylinders CYL_i, $i > 0$. Only the maximal indexed cylinders of these has infinite intersection with L, and has some special properties. This allows identification as long as one knows r and is allowed one further superset query. This r can be obtained using one superset query, where least counterexample is presented. However this r cannot be obtained using (bounded number of) arbitrary counterexamples to superset queries, thus making it difficult to identify $\mathcal{L}$. We now proceed formally.

Claim 30. $\mathcal{L} \in \mathbf{LSupQ}^2\mathbf{Ex}$.

Proof. Suppose T is a text for $L \in \mathcal{L}$. We first describe the two queries that the learner will make. First query is whether $N - \mathrm{CYL}_0$ is a superset of the input language. As no language in $\mathcal{L}$ is contained in $N - \mathrm{CYL}_0$, one will get a least counterexample. Suppose this counterexample is $\langle 0, 3^{n+2}+2-r\rangle$ (note that this r would correspond to r as in the definition of $\mathcal{L}$). Then, on any input $T[s]$, compute $S = \{i > 0 \mid \mathrm{content}(T[s]) \cap \mathrm{CYL}_i \neq \emptyset\}$. If S contains at least r elements, then let $e = \max(S)$ and query whether $(N - \mathrm{CYL}_e) \cup \{\langle e, x\rangle \mid x \in W_e\})$ is a superset of the input language. (Note that for languages in $\mathcal{L}$, the above set S would contain exactly r elements).

If the answer is yes, then learner outputs in the limit on T a grammar for: $[\mathrm{content}(T) \cap (N - \mathrm{CYL}_e)] \cup \{\langle e, x\rangle \mid x \in W_e\}$.

Otherwise the learner computes, in the limit, the least w such that both $\langle e, 2w\rangle$ and $\langle e, 2w+1\rangle$ do not belong to L (if $L \in \mathcal{L}$, then there must exist such a w). Then, the learner outputs, in the limit, a grammar for $[\mathrm{content}(T) \cap (N - \mathrm{CYL}_e)] \cup \{\langle e, x\rangle \mid x < 2w, \langle e, x\rangle \in \mathrm{content}(T)\} \cup \{\langle e, 2x\rangle \mid x > w\}$.

It is easy to verify that the above learner would $\mathbf{LSupQ}^2\mathbf{Ex}$-identify $\mathcal{L}$.

Claim 31. *(a) For all* $t \in N$, $\mathcal{L} \notin \mathbf{SupQ}^n\mathbf{Bc}^t$.
(b) $\mathcal{L} \notin \mathbf{SupQ}^n\mathbf{Ex}^*$.

Proof. We only show part (a). Part (b) can be shown using Corollary 21 instead of using Lemma 20.

Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{SupQ}^n\mathbf{Bc}^t$-identifies $\mathcal{L}$. We will maintain two variables, l_i and u_i. Intuitively, it will be the case that we have flexibility to choose any r, with $l_i \leqslant 3^{n+2}+2-r \leqslant u_i$, for r as in the definition of $\mathcal{L}$. Additionally, we will also define R_i and σ_i. Initially, $R_0 = \emptyset$ and $l_0 = 1, u_0 = 3^{n+2}+1$, and σ_0 contain only $\langle 0, u_0\rangle$.

Inductively define $\sigma_{i+1}, R_{i+1}, l_{i+1}, u_{i+1}$, for $i < n$ as follows.
(* The construction is non-effective. *)
(* By induction, we will have the following invariants:
(a) $(u_i - l_i) = 3^{n+2-i}$.
(* Note in particular that $(u_i - l_i) \geqslant 3$. *)
(b) $R_i = \{j > 0 \mid \text{content}(\sigma_i) \cap \text{CYL}_j \neq \emptyset\}$.
(c) $\text{card}(R_i) = 3^{n+2} + 1 - u_i$.
(d) $\mathbf{M}$ has already asked i queries on proper prefixes of σ_i.
(e) $\{\langle 0, x\rangle \mid u_i \leqslant x \leqslant 3^{n+2}+1\} = \text{content}(\sigma_i) \cap \text{CYL}_0$.
(f) Answers given to $\mathbf{M}$ on queries made on proper prefixes of σ_i are consistent with any language L such that $\text{content}(\sigma_i) \subseteq L \subseteq N - \{\langle 0, x\rangle \mid x < l_i \text{ or } x > 3^{n+2}+1\}$.*)

1. Check if there exists a $\sigma \supseteq \sigma_i$ such that, for some $e \notin R_i \cup \{0\}$, $\text{content}(\sigma) \subseteq \{\langle 0, x\rangle \mid u_i \leqslant x \leqslant 3^{n+2}+1\} \cup \bigcup_{j \in R_i \cup \{e\}} \text{CYL}_e$ and $\mathbf{M}$ asks a query on σ.
   If there is no such σ, then σ_j, R_j, l_j, u_j for $j > i$ do not get defined.
   If there exists such a σ, then fix one such σ and corresponding e and proceed as follows.
   (* Note that we will have $u_{i+1} < u_i$, as we need to have $R_{i+1} \supseteq R_i \cup \{e\}$. *)
2. Suppose the queried language is Q.
3. If $Q \supseteq N - \{\langle 0, x\rangle \mid x > 3^{n+2}+1\}$, then answer the query as yes.
   Let $u_{i+1} = l_i + \frac{u_i - l_i}{3}$, and $l_{i+1} = l_i$.
   Let $R_{i+1} \supseteq R_i \cup \{e\}$, be such that
   R_{i+1} contains exactly $3^{n+2} + 1 - u_{i+1}$ elements and
   R_{i+1} does not contain 0.
   Let σ_{i+1} be an extension of σ such that
   $R_{i+1} = \{j > 0 \mid \text{content}(\sigma_{i+1}) \cap \text{CYL}_j \neq \emptyset\}$, and
   $\text{content}(\sigma_{i+1}) \cap \text{CYL}_0 = \{\langle 0, x\rangle \mid u_{i+1} \leqslant x \leqslant 3^{n+2}+1\}$.
4. If $Q \not\supseteq N - \{\langle 0, x\rangle \mid x > 3^{n+2}+1\}$, then we consider the following cases:

4.1 Q misses out an element in $(N - \text{CYL}_0) \cup \{\langle 0, x\rangle \mid l_i + \frac{u_i - l_i}{3} \leqslant x \leqslant 3^{n+2}+1\}$.
   Then let $\langle e', y\rangle$ be one such element.
   Answer the query as no, with $\langle e', y\rangle$ as the counterexample.
   Let $u_{i+1} = l_i + \frac{u_i - l_i}{3}$, and $l_{i+1} = l_i$.
   Let $R_{i+1} \supseteq R_i \cup \{e\}$ be such that
   R_{i+1} contains exactly $3^{n+2} + 1 - u_{i+1}$ elements,
   R_{i+1} does not contain 0, and
   if $e' \neq 0$, then $e' \in R_{i+1}$.
   Let σ_{i+1} be an extension of σ such that
   $R_{i+1} = \{j > 0 \mid \text{content}(\sigma_{i+1}) \cap \text{CYL}_j) \neq \emptyset\}$,
   $\langle e', y\rangle \in \text{content}(\sigma_{i+1})$ and
   $\text{content}(\sigma_{i+1}) \cap \text{CYL}_0 = \{\langle 0, x\rangle \mid u_{i+1} \leqslant x \leqslant 3^{n+2}+1\}$.

4.2 $Q \supseteq (N - \text{CYL}_0) \cup \{\langle 0, x\rangle \mid l_i + \frac{u_i - l_i}{3} \leqslant x \leqslant 3^{n+2}+1\}$.

Then, answer the query as yes.
Let $l_{i+1} = l_i + \frac{u_i - l_i}{3}$, and $u_{i+1} = l_i + \frac{2(u_i - l_i)}{3}$, and
Let $R_{i+1} \supseteq R_i \cup \{e\}$ be such that
R_{i+1} contains exactly $3^{n+2} + 1 - u_{i+1}$ elements and
R_{i+1} does not contain 0.
Let σ_{i+1} be an extension of σ such that
$R_{i+1} = \{j > 0 \mid \text{content}(\sigma_{i+1}) \cap \text{CYL}_j) \neq \emptyset\}$, and
$\text{content}(\sigma_{i+1}) \cap \text{CYL}_0 = \{\langle 0, x\rangle \mid u_{i+1} \leqslant x \leqslant 3^{n+2} - 1\}$.
(* We assume without loss of generality that if $\sigma \subset \sigma' \subseteq \sigma_{i+1}$, then $\mathbf{M}$ does not ask any questions. If not, then one can just delay these questions beyond σ_{i+1}, without effecting this construction. *)
End

It is easy to verify that the invariants are satisfied. Let m be the largest number such that σ_m gets defined. Note that $\mathbf{M}$ does not ask any more questions on any text T such that $\sigma_m \subseteq T$, and $\text{content}(T) \subseteq \{\langle 0, x\rangle \mid u_m \leqslant x \leqslant 3^{n+2} + 1\} \cup \bigcup_{j \in R_m \cup \{e\}} \text{CYL}_j$, for any fixed $e > 0$. Thus, one can design a machine $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{TxtBc}^t$-identifies all L such that $\mathbf{M}$ $\mathbf{SupQ}^n\mathbf{Bc}^t$ identifies L and $\text{content}(\sigma_m) \subseteq L \subseteq \{\langle 0, x\rangle \mid u_m \leqslant x \leqslant 3^{n+2} + 1\} \cup \bigcup_{j \in R_m \cup \{e\}} \text{CYL}_j$, for some e.

Now, let F be as in Lemma 20. By Kleene's recursion theorem [28], there exists an $e > \max(R_m)$, such that $W_e = W_{F(e, \text{content}(\sigma_m), \mathbf{M}')}$. It now follows from Lemma 20 that $\mathbf{M}'$ does not $\mathbf{TxtBc}^t$-identify some language $L \in \mathcal{L}$, such that $\text{content}(\sigma_m) \subseteq L$. Thus, $\mathbf{M}$ does not $\mathbf{SupQ}^n\mathbf{Bc}^t$-identify L and hence $\mathcal{L}$. □

For learning with a bounded number of negative counterexamples to conjectures, advantage of having least counterexample is slightly complicated. Roughly speaking, one can simulate the effect of using the least counterexamples by doubling the number of negative answers in the restricted type of this model when the learner gets only the answer "no" if the current conjecture is not a subset of the target language.

Theorem 32. *Suppose* $a \in N \cup \{*\}$, $n \in N$, $\mathbf{I} \in \{\mathbf{Ex}^a, \mathbf{Bc}^a\}$.
$\mathbf{LNC}^n\mathbf{I} \subseteq \mathbf{ResNC}^{2n-1}\mathbf{I}$.

Proof. We first show that $\mathbf{LNC}^n\mathbf{I} \subseteq \mathbf{ResNC}^{2n}\mathbf{I}$. We will then explain how one counterexample can be saved.

Suppose $\mathbf{M}$ $\mathbf{LNC}^n\mathbf{I}$-identifies $\mathcal{L}$. Then $\mathbf{M}'$ simulates $\mathbf{M}$, outputting the conjectures of $\mathbf{M}$. If a conjecture j of $\mathbf{M}$ gets a no answer (i.e., $W_j \not\subseteq$ input language), then $\mathbf{M}'$ also outputs grammars for $W_j \cap \{y\}$, in increasing order of y, until a no answer is received. Then $\mathbf{M}'$ passes this y (i.e., the least y such that $W_j \cap \{y\}$ generates a no answer) to $\mathbf{M}$ as a counterexample, and proceeds with the simulation.

It is easy to verify that the number of counterexamples received by $\mathbf{M}'$ is exactly the double of the number of counterexamples given to $\mathbf{M}$ during the simulation.

To save one "no" answer, do the simulation as above, except that after $\mathbf{M}'$ receives the $(2n-1)$-th no answer (that is we need to provide $\mathbf{M}$ with the n-th counterexample), proceed as in the proof of $\mathbf{LNC}^1\mathbf{I} \subseteq \mathbf{ResNC}^1\mathbf{I}$ from Theorem 25 to get the counterexample for the latest conjecture of $\mathbf{M}$.

Theorem follows. □

Now we show that the bound $(2n-1)$ on the number of negative answers in the restricted **NC**-model needed to simulate n least counterexamples to conjectures is tight: $(2n-2)$ "no" answers (with counterexamples) are not enough.

Theorem 33. *Suppose* $n \geqslant 1$. $\mathbf{LNC}^n\mathbf{Ex} - \mathbf{NC}^{2n-2}\mathbf{Bc}^* \neq \emptyset$.

Proof. Recall that D_k is the k-th finite set.

Let

$L_{i,k} = \{\langle i,k,x\rangle \mid x \in N\}$.
$X_i = L_{i,0}$.
$Y_i^j = \{\langle i,0,x\rangle \mid x < 3j\} \cup L_{i,j+1}$.
$Z_i^{j,k} = \{\langle i,0,x\rangle \mid x < 3j+1\} \cup \{\langle i,j+1,x\rangle \mid x \leqslant k\}$.
$U_i^j = \{\langle i,0,x\rangle \mid x < 3j+2\}$.
$\mathcal{L}_i = \{X_i\} \cup \{Y_i^j \mid j \in N\} \cup \{U_i^j \mid j \in N\} \cup \{Z_i^{j,k} \mid j,k \in N\}$.
$\mathcal{C}_n = \{L \mid (\exists A \mid \mathrm{card}(A) \leqslant n)[L$ is formed by picking one language from each $\mathcal{L}_i, i \in A$, and then taking the union$]\}$.

Intuitively, each $L \in \mathcal{L}_i$ is either X_i or an initial segment of X_i, and the least such element from $X_i - L$, indicates the form of L (i.e., whether it is Y_i^j, $Z_i^{j,k}$ or U_i^j, for some j,k). This allows for easy learnability when one gets n least counterexamples. However, it will be shown below that $(2n-2)$ negative answers are not enough for learning the above class.

Claim 34. $\mathcal{C}_n \in \mathbf{LNC}^n\mathbf{Ex}$.

Proof. A learner can $\mathbf{LNC}^n\mathbf{Ex}$-identify the class $\mathcal{C}_n$ as follows. On input (σ,σ'), do as follows.

Let $A = \{i \mid (\exists x,y)[\langle i,x,y\rangle \in \mathrm{content}(\sigma)]\}$. Let $A' = \{i \mid (\exists j)[\langle i,0,3j\rangle \in \mathrm{content}(\sigma')]\}$. Let $A'' = \{i \mid (\exists j)[\langle i,0,3j+1\rangle \in \mathrm{content}(\sigma')$ or $\langle i,0,3j+2\rangle \in \mathrm{content}(\sigma')]\}$.

It would be the case that for input from $\mathcal{C}_n$ the sets A', A'' are disjoint subsets of A (see below). For $i \in A'$, let j_i be such that $\langle i,0,3j_i\rangle \in \mathrm{content}(\sigma')$.

Output a (standard) grammar for the language:

$$\bigcup_{i \in A-A'-A''} X_i$$

$$\cup \bigcup_{i \in A'} Y_i^{j_i}$$

$$\cup \bigcup_{i \in A''} \mathrm{content}(\sigma)$$

Now consider any input language $L \in \mathcal{C}_n$. By induction we claim that counterexamples received would only be of the form $\langle i,0,z\rangle$. Furthermore, for the same i, these counterexamples may only appear on conjectures output by the learner on inputs of form $(\sigma = \tau\diamond\langle i,x,y\rangle,\sigma')$, where τ does not contain any element of form $\langle i,x',y'\rangle$, and σ' is the sequence of counterexamples/# obtained

based on earlier conjectures (thus in particular, there would be at most one counterexample of form $\langle i, 0, z\rangle$, for any given i, that the learner will receive—ensuring that A', A'' are disjoint as claimed earlier).

Now, consider any i such that the input language L contains a language from $\mathcal{L}_i$ as its subset. The first time an element of form $\langle i, x, y\rangle$, for the given i, appears in the input, X_i would be included in the conjectured language. We consider the following cases.

Case 1: There is no counterexample to this conjecture.

In this case the language from $\mathcal{L}_i$, which is a subset of L, must be X_i. Furthermore, for any future input, we will never have a counterexample of form $\langle i, x, y\rangle$, and thus i will never be placed in A', A''. Thus, X_i would be contained in the conjectured language.

Case 2: There is a counterexample of form $\langle i, 0, 3j\rangle$.

In this case the language from $\mathcal{L}_i$ which is a subset of L must be Y_i^j. Also, i will be placed in A'. Furthermore, we will never have a counterexample of form $\langle i, x, y\rangle$, for any future input. Thus, Y_i^j would be contained in the conjectured language.

Case 3: There is a counterexample of form $\langle i, 0, 3j+1\rangle$ or $\langle i, 0, 3j+2\rangle$.

In this case the language from $\mathcal{L}_i$, which is a subset of L, must be finite. Also, i will be placed in A''. Furthermore, we will never have a counterexample of form $\langle i, x, y\rangle$, for any future input, due to the form of conjectures made by the learner.

From the above cases, it is easy to verify that induction hypothesis would be satisfied, and eventually the learner would converge to a grammar for L. Thus, $\mathcal{C}_n \in \mathbf{LNC}^n\mathbf{Ex}$.

Claim 35. $\mathcal{C}_n \notin \mathbf{NC}^{2n-2}\mathbf{Bc}^*$.

Proof. Suppose by way of contradiction $\mathbf{M}$ $\mathbf{NC}^{2n-2}\mathbf{Bc}^*$-identifies $\mathcal{L}$.

Initially, let $\sigma_0 = \Lambda$, $\sigma'_0 = \Lambda$. Intuitively, σ'_s would denote the sequence of counterexamples/# provided to $\mathbf{M}$ on input σ_s. Let $A_0 = S_0 = \emptyset$. Intuitively, $A = \bigcup A_s$ plus (one more element) would mimic the A as in the definition of $\mathcal{C}_n$. S_s would denote the set of elements we have decided not to be in A (elements of S_s represent the spoiled classes, due to some counterexamples used). As we build up the set A, we would also freeze the languages $F_r \in \mathcal{L}_r$, for $r \in A_s$, such that $F_r \subseteq L$, the diagonalizing language being constructed.

For $s \leqslant n-2$, inductively define σ_{s+1}, σ'_{s+1}, A_{s+1}, S_{s+1}, and F_r for $r \in A_{s+1}$, as follows.

(* The construction is non-effective. *)

(* Following invariants will be satisfied:

(a) $A_s \cap S_s \neq \emptyset$.

(b) $\mathrm{card}(A_s) = s$. S_s is finite.

(c) $\mathrm{content}(\sigma_s) \subseteq \bigcup_{r\in A_s} F_r$.

(d) For $r \in A_s$, $F_r \in \mathcal{L}_r$.

(e) Counterexamples/Answers given to $\mathbf{M}$ via σ' are consistent with any language L such that

$\bigcup_{r\in A_s} F_r \subseteq L \subseteq \bigcup_{r\in A_s} F_r \cup \bigcup_{r\notin S_s\cup A_s}\{\langle r, x, y\rangle \mid x, y \in N\}$.

*)

1. Let i be a member of $N - (S_s \cup A_s)$.
2. If there exists a $\sigma \supseteq \sigma_s$ such that $\mathrm{content}(\sigma) \subseteq X_i \cup \bigcup_{r\in A_s} F_r$, and one of the following is satisfied:

2.1. $W_{\mathbf{M}(\sigma,\sigma'_s\#^{|\sigma|-|\sigma_s|})} - (X_i \cup \bigcup_{r\in A_s} F_r) \neq \emptyset$.
2.2. Not 2.1, and $W_{\mathbf{M}(\sigma,\sigma'_s\#^{|\sigma|-|\sigma_s|})} \cap X_i$ is infinite.

3. Then, pick smallest such σ (we will argue below that there must exist such a σ).
Pick j such that $\langle i,0,3j\rangle > \max(X_i \cap (\text{content}(\sigma) \cup \bigcup_{\gamma:\sigma_s\subseteq\gamma\subset\sigma} W_{\mathbf{M}(\gamma,\sigma'_s\#^{|\gamma|-|\sigma_s|})}))$.
(* Note that, for any $\gamma, \sigma_s \subseteq \gamma \subset \sigma$, as 2.2 did not succeed, $X_i \cap W_{\mathbf{M}(\gamma,\sigma'_s\#^{|\gamma|-|\sigma_s|})}$ must be finite. Thus, such a j exists. *)

4. If 2.1 holds:
Let $\langle i',j',k'\rangle$ be an element of $W_{\mathbf{M}(\sigma,\sigma'_s\#^{|\sigma|-|\sigma_s|})} - (X_i \cup \bigcup_{r\in A_s} F_r)$.
If 2.2 holds:
Let $\langle i',j',k'\rangle = \langle i,0,k'\rangle$, where $k' \geqslant 3j+3$ and $\langle i,0,k'\rangle \in W_{\mathbf{M}(\sigma,\sigma'_s\#^{|\sigma|-|\sigma_s|})} \cap X_i$.

5. Let $\tau = \sigma\#$ and $\tau' = \sigma_s\#^{|\sigma|-|\sigma_s|}\langle i',j',k'\rangle$.
(* That is we give counterexample $\langle i',j',k'\rangle$ to $W_{\mathbf{M}(\sigma,\sigma'_s\#^{|\sigma|-|\sigma_s|})}$. *)
If there exists a $\sigma \supseteq \tau$ such that $\text{content}(\sigma) \subseteq Y_i^j \cup \bigcup_{r\in A_s} F_r$, and $W_{\mathbf{M}(\sigma,\tau'\#^{|\sigma|-|\tau|})}$ contains an element of form $\langle i'',j'',k''\rangle$ such that one of the following conditions is satisfied:
5.1. $\langle i'',j'',k''\rangle \notin Y_i^j \cup \bigcup_{r\in A_s} F_r$,
5.2. Not 5.1 and $i = i''$, $j'' = j+1$, and $\langle i'',j'',k''\rangle \notin \text{content}(\sigma)$.

6. Then, pick a shortest such σ (we will argue below that there must exists such a σ).
If 5.1 holds,
Let $F_i = Y_i^j$.
Let $\sigma_{s+1} = \sigma\#$ and $\sigma'_{s+1} = \tau'\#^{|\sigma|-|\tau|}\langle i'',j'',k''\rangle$.
If 5.2 holds,
Let $F_i = Z_i^{j,k}$, for $k = \max(\{x \mid \langle i,j+1,x\rangle \in \text{content}(\sigma)\})$.
Let $\sigma_{s+1} = \sigma\#$ and $\sigma'_{s+1} = \tau'\#^{|\sigma|-|\tau|}\langle i'',j'',k''\rangle$.
(* Note that we give counterexample $\langle i'',j'',k''\rangle$ to $W_{\mathbf{M}(\sigma,\tau'\#^{|\sigma|-|\tau|})}$. *)

7. Let $A_{s+1} = A_s \cup \{i\}$.
Let $S_{s+1} = S_s \cup [\{i',i''\} - (A_s \cup \{i\})]$.

End

It is easy to verify that the invariants are maintained by the construction. Specially note that the invariant (e) is maintained as any conjecture of $\mathbf{M}$ on positive input data γ, with $\sigma_s \subseteq \gamma \subset \sigma_{s+1}$, which did not get a negative counterexample, indeed enumerates a subset of $F_i \cup \bigcup_{r\in A_s} F_r$. (Note that based on the definition of j, we included the elements in $X_i \cap (\text{content}(\sigma) \cup \bigcup_{\gamma:\sigma_s\subseteq\gamma\subset\sigma} W_{\mathbf{M}(\gamma,\sigma'_s\#^{|\gamma|-|\sigma_s|})})$ into F_i by choosing an appropriate j at step 3. Similarly, k is chosen appropriately in step 6, if 5.2 holds).

We first claim that the above construction finishes for every $s \leqslant n-2$ (i.e., $\sigma_{n-1}, \sigma'_{n-1}$ get defined). If not, then let s be least such that σ_s, σ'_s get defined but $\sigma_{s+1}, \sigma'_{s+1}$ do not. Now consider the construction above while trying to define $\sigma_{s+1}, \sigma'_{s+1}$.

If the "If" statement at step 2 does not hold, then $\mathbf{M}$ does not $\mathbf{NC}^{2n-2}\mathbf{Bc}^*$-identify the language $X_i \cup \bigcup_{r\in A_s} F_r$, which is a member of $\mathcal{C}_n$ (as 2.1/2.2 do not hold for any σ extending σ_s, and $\text{content}(\sigma) \subseteq X_i \cup \bigcup_{r\in A_s} F_r$).

If the "If" statement at step 5 does not hold, then $\mathbf{M}$ does not $\mathbf{NC}^{2n-2}\mathbf{Bc}^*$-identify the language $Y_i^j \cup \bigcup_{r\in A_s} F_r$, which is a member of $\mathcal{C}_n$ (as 5.1/5.2 do not hold for any $\langle i',j',k'\rangle$ enumerated by $W_{\mathbf{M}(\sigma,\tau'\#^{|\sigma|-|\tau|})}$, for any σ extending τ, and $\text{content}(\sigma) \subseteq Y_i^j \cup \bigcup_{r\in A_s} F_r$).

Thus, $\sigma_{n-1}, \sigma'_{n-1}$ must get defined. Now, on the input $(\sigma_{n-1}, \sigma'_{n-1})$, $\mathbf{M}$ has already received $2n-2$ negative counterexamples (2 counterexamples each during the definition of σ_{i+1}, for $i < n-1$). Let $i \in N - (A_{n-1} \cup S_{n-1})$. Now, $\mathbf{M}$ needs to $\mathbf{NC}^{2n-2}\mathbf{Bc}^*$-identify $F_i \cup \bigcup_{r \in A_{n-1}} F_r$, for every possible $F_i \in \mathcal{L}_i$, without receiving any more counterexamples. This is impossible, as no machine can $\mathbf{TxtBc}^*$-identify $X_i \cup \bigcup_{r \in A_{n-1}} F_r$, and $U_i^j \cup \bigcup_{r \in A_{n-1}} F_r$, for all j, by Proposition 11. □

6.3. Queries with arbitrary counterexamples versus restricted queries

We now consider the advantage of having arbitrary counterexamples versus being just told that there exists a counterexample. Again we separate the result for superset queries from the others. Also, due to Theorem 32, for learning via negative counterexamples to conjectures, only a limited version of the hierarchy can exist.

First, we show that there exists a class of languages that can be $\mathbf{Ex}$-learned using just two subset or equivalence queries returning arbitrary counterexamples, but cannot be learned by any $\mathbf{Bc}^*$-learner via any m restricted queries of either type. For $\mathbf{NC}$-learners, a class of the same style is used to demonstrate that an $\mathbf{Ex}$-learner getting n arbitrary counterexamples can do better than any $\mathbf{Bc}^*$-learner getting at most $(2n-2)$ counterexamples.

Theorem 36. *Suppose $n, m \in N$.*
Let $\mathcal{L}_n = \{N - \bigcup_{i \in A} CYL_i \mid \mathrm{card}(A) \leqslant n\}$. Then,
(a) $\mathcal{L}_n \in \mathbf{EquQ}^n\mathbf{Ex} \cap \mathbf{SubQ}^n\mathbf{Ex} \cap \mathbf{NC}^n\mathbf{Ex}$.
(b) For all n, $\mathcal{L}_2 \notin \mathbf{ResEquQ}^n\mathbf{Bc}^* \cup \mathbf{ResSubQ}^n\mathbf{Bc}^*$.
(c) For $n \geqslant 1$, $\mathcal{L}_n \notin \mathbf{ResNC}^{2n-2}\mathbf{Bc}^*$.

Proof. (a) Fix n. We first define a learner which $\mathbf{EquQ}^n\mathbf{Ex}$-identifies ($\mathbf{SubQ}^n\mathbf{Ex}$-identifies) $\mathcal{L}_n$. This learner works for both equivalence or subset queries.

The learner asks n queries as follows. Let X_r denote the set of (negative) counterexamples received before the r-th query is made (for $j = 1$, $X_j = \emptyset$). Let $S_r = \{j \mid \langle j, x \rangle \in X_r\}$. Then ask an (equivalence/subset) query for the language $N - \bigcup_{i \in S_r} \mathrm{CYL}_i$. Note that all the counterexamples received by the learner would always be negative as long as the input is a language from the class $\mathcal{L}_n$.

After asking the n-queries as above, let X_{n+1} denote the set of negative counterexamples received for the n queries. Let $S_{n+1} = \{j \mid \langle j, x \rangle \in X_{n+1}\}$. Then output a grammar for $N - \bigcup_{i \in S_{n+1}} \mathrm{CYL}_i$.

Note that if all the queries receive a negative counterexample, then S_{n+1} must be of size n, and $N - \bigcup_{i \in S_{n+1}} \mathrm{CYL}_i$, must be the input language. On the other hand, if some query (say r-th query) does not receive a counterexample, then the input language must be $N - \bigcup_{i \in S_r} \mathrm{CYL}_i$, and no further counterexamples are received by the learner. Thus, again, $N - \bigcup_{i \in S_{n+1}} \mathrm{CYL}_i$ must be the input language.

Thus, the learner indeed $\mathbf{EquQ}^n\mathbf{Ex}$-identifies ($\mathbf{SubQ}^n\mathbf{Ex}$-identifies) $\mathcal{L}_n$.

For $\mathbf{NC}^n\mathbf{Ex}$-identification, instead of querying about languages as above, we just conjecture the corresponding language. If the input language is not the conjectured language, then it will eventually get a counterexample. Then we can proceed as above.

(b) Suppose by way of contradiction that $\mathbf{M}$ witnesses that $\mathcal{L}_2 \in \mathbf{ResSubQ}^n\mathbf{Bc}^*$ ($\mathbf{ResEquQ}^n\mathbf{Bc}^*$). We prove a stronger result, where we allow the machine to ask both subset or equivalence queries,

as long as there are at most n queries in total (we assume without loss of generality that $\mathbf{M}$ does not ask more than n queries on any text, even if the text is for a language not in $\mathcal{L}_2$ or if the answers are wrong).

We will use the variable S_i. Intuitively, S_i denotes that members of A in the definition of diagonalizing language in $\mathcal{L}_2$ can be chosen from S_i. As long as A is non-empty, we will have that this would be consistent with all the answers provided so far. Initially let $\sigma_0 = \Lambda$. Let $S_0 = N$.

Inductively define σ_{i+1}, S_{i+1}, for $i < n$ as follows.
(* The construction is non-effective. *)
(* Following invariants will be satisfied.
(a) $\text{content}(\sigma_i) \subseteq N - \bigcup_{j \in S_i} \text{CYL}_j$.
(b) $\text{card}(S_i) = \infty$.
(c) $\mathbf{M}$ has already asked i questions on proper prefixes of σ_i.
(d) Answers given to $\mathbf{M}$ on questions are consistent with any language L such that $\bigcup_{j \notin S_i} \text{CYL}_j \subseteq L$, as long as $L \neq N$ and $L \in \mathcal{L}_2$.
*)
1. Check if there exists a $\sigma \supseteq \sigma_i$ such that $\mathbf{M}$ on σ asks a query.
If there is no such σ, then $\sigma_{i'}$, $i' > i$ do not get defined.
If there exists such a σ, then choose a shortest such σ, and proceed as follows.
2. Let $\sigma_{i+1} = \sigma\#$. Let Q be the queried language.
Define S_{i+1} based on following cases.

2.1 $\mathbf{M}$ asks an equivalence query on σ.
Answer the query no.
If $Q \neq N$,
Then pick an element $\langle j, x \rangle$ missing from Q.
Let $S_{i+1} = S_i - (\{j\} \cup \{r \mid \text{content}(\sigma) \cap \text{CYL}_r \neq \emptyset\})$.
Else let $S_{i+1} = S_i - \{r \mid \text{content}(\sigma) \cap \text{CYL}_r \neq \emptyset\}$.

2.2 $\mathbf{M}$ asks a subset query on σ, and $S_i \cap \{r \mid \text{CYL}_r \cap W_{\mathbf{M}(\sigma)} \neq \emptyset\}$ is finite.
Answer the query yes.
Let $S_{i+1} = S_i - (\{r \mid \text{content}(\sigma) \cap \text{CYL}_r \neq \emptyset\} \cup \{r \mid \text{CYL}_r \cap W_{\mathbf{M}(\sigma)} \neq \emptyset\})$.

2.3 $\mathbf{M}$ asks a subset query on σ, and $S_i \cap \{r \mid \text{CYL}_r \cap W_{\mathbf{M}(\sigma)} \neq \emptyset\}$ is infinite.
Answer the query no.
Let $S_{i+1} = [S_i - \{r \mid \text{content}(\sigma) \cap \text{CYL}_r \neq \emptyset\}] \cap \{r \mid \text{CYL}_r \cap W_{\mathbf{M}(\sigma)} \neq \emptyset\}$.
(* Note that all the answers given above are consistent with choosing elements of A, in the definition of $\mathcal{L}_2$, from S_{i+1} as long as A is non-empty. *)
End

It is easy to verify that the invariants are maintained by the construction.

Let m be largest number such that σ_m is defined. Note that $\mathbf{M}$ does not make any more queries on $\sigma \supseteq \sigma_m$. Let $i \in S_m$. Now, $\mathbf{M}$ must $\mathbf{Bc}^*$-identify $N - \text{CYL}_i$, as well as $N - (\text{CYL}_i \cup \text{CYL}_j)$, for all $j \in S_m$, without asking any more queries beyond σ_m. This is not possible by Proposition 12.

(c) Fix n. Suppose by way of contradiction that $\mathbf{MResNC}^{2n-2}\mathbf{Bc}^*$-identifies $\mathcal{L}_n$.

For ease of writing the proof, we will not provide the second text to $\mathbf{M}$, but only mention which conjectures get no answer (to the question, whether the conjecture is subset of the input language). All other conjectures in the construction are supposed to get yes answer.

We will use variables S_i and R_i. Intuitively, S_i denotes that members of A in the definition of diagonalizing language in $\mathcal{L}_n$ can be chosen from S_i. Members of R_i are committed to be in A. We will have that this would be consistent with all the answers provided so far. Initially let $\sigma_0 = \Lambda$. Let $S_0 = N, R_0 = \emptyset$.

Inductively define $\sigma_{i+1}, S_{i+1}, R_{i+1}$, for $i < n-1$ as follows.
(* The construction is non-effective. *)
(* We will maintain the following invariants:
(a) $\mathrm{card}(S_i) = \infty$.
(b) $\mathrm{card}(R_i) = i$.
(c) $\mathbf{M}$ has already received $2i$ no answers to its conjectures.
(d) All the answers given to machine $\mathbf{M}$ on proper prefixes of σ_i are consistent with the input language being any L which satisfies: $(N - \bigcup_{x \in S_i \cup R_i} \mathrm{CYL}_x) \subseteq L \subseteq (N - \bigcup_{x \in R_i} \mathrm{CYL}_x)$.
*)

1. Check if there exists a $\sigma \supseteq \sigma_i$ such that
   (i) $\mathrm{content}(\sigma) \subseteq N - \bigcup_{j \in R_i} \mathrm{CYL}_j$, and
   (ii) On proper prefixes of σ, one answers "no" only to conjectures which include an element from $\bigcup_{j \in R_i} \mathrm{CYL}_j$, and
   (iii) $\mathbf{M}$ on σ conjectures a language W_r which contains elements from CYL_w for infinitely many $w \in S_i$.
   If there is no such σ, then $\sigma_{i'}, i' > i$ do not get defined.
   If there exists such a σ, then choose a shortest such σ, and proceed as follows.
2. Answer the latest conjecture as no.
   Let $Z = (S_i - (\{j \mid \mathrm{content}(\sigma) \cap \mathrm{CYL}_j \neq \emptyset\} \cup \bigcup_{\sigma_i \subseteq \sigma'' \subset \sigma} \{j \mid \mathrm{CYL}_j \cap W_{\mathbf{M}(\sigma'')} \neq \emptyset\})) \cap \{j \mid W_r \cap \mathrm{CYL}_j \neq \emptyset\}$.
3. Check if there exists a $\sigma' \supseteq \sigma\#$ such that
   (iv) $\mathrm{content}(\sigma) \subseteq N - \bigcup_{j \in R_i} \mathrm{CYL}_j$, and
   (v) on proper prefixes of σ', one answers "no" only to conjectures which include an element from $\bigcup_{j \in R_i} \mathrm{CYL}_j$, and
   (vi) $\mathbf{M}$ on σ' conjectures a language $W_{r'}$ such that $W_{r'}$ contains an element from $\mathrm{CYL}_{w'}$ for some $w' \in Z - \{j \mid \mathrm{content}(\sigma') \cap \mathrm{CYL}_j \neq \emptyset\}$.
   If there is no such σ', then $\sigma_{i'}, i' > i$ do not get defined.
   If there exists such a σ', then choose a shortest such σ', and proceed as follows.
4. Answer no to this conjecture.
   Let $R_{i+1} = R_i \cup \{w'\}$.
   Let $S_{i+1} = Z - (\{w'\} \cup \{j \mid \mathrm{content}(\sigma') \cap \mathrm{CYL}_j \neq \emptyset\})$.
   Let $\sigma_{i+1} = \sigma'\#$.

End

It is easy to verify that the invariants are maintained by the construction.

Let m be largest such that σ_m is defined.

If $m = n - 1$, then we have already answered $2n - 2$ questions negatively (two for each construction of σ_{i+1} from σ_i), and thus $\mathbf{M}$ needs to $\mathbf{Bc}^*$-identify $N - \bigcup_{j \in R_m} \mathrm{CYL}_j$, as well as $(N - \bigcup_{j \in R_m} \mathrm{CYL}_j) - \mathrm{CYL}_{j'}$ for all $j' \in S_m$. An impossible task by Proposition 12.

Otherwise $m < n - 1$. We consider two cases.

Case 1: In trying to define σ_{m+1}, we are not able to find σ as above.

In this case $\mathbf{M}$ does not $\mathbf{Bc}^*$-identify the language $L = N - \bigcup_{j \in R_m} \mathrm{CYL}_j$, on any text for L which extends σ_i, as $\mathbf{M}$ does not output a grammar for L on any part of the text extending σ.

Case 2: In trying to define σ_{m+1}, we are able to find σ but not σ' as above.

Note that when defining σ, we had given no answer at the conjecture of $\mathbf{M}$ on σ. Thus, we had committed that "there is at least one more missing cylinder beyond R_m" (plus this missing cylinder to come from Z).

So, let $p \in Z$. Now we claim that $\mathbf{M}$ does not $\mathbf{Bc}^*$-identify the language $L = N - (\mathrm{CYL}_p \cup \bigcup_{j \in R_m} \mathrm{CYL}_j)$. Note that on any text T for L, with $\sigma \subseteq T$, $\mathbf{M}$ does not output a grammar for (finite variant) of L. Further note that the "answers" given to $\mathbf{M}$'s conjectures on T beyond σ are no iff the conjecture contains an element of $\bigcup_{x \in R_m} \mathrm{CYL}_x$. These answers are consistent with the defined L, as $\mathbf{M}$ does not output a grammar (beyond σ) containing any element of CYL_p, as the search for σ' did not succeed. Thus, $\mathbf{M}$ does not $\mathbf{ResNCBc}^*$-identify L.

From the above cases we have that $\mathbf{M}$ does not $\mathbf{NC}^{2n-2}\mathbf{Bc}^*$-identify $\mathcal{L}_n$. □

Our next theorem shows that **Ex**-learners using just two superset queries and getting arbitrary counterexamples can sometimes do better than any $\mathbf{Bc}^t$-learner ($t \in N$) using any n number of restricted superset queries. Note that this result cannot be generalized for diagonalization against $\mathbf{ResSupQ}^n\mathbf{Bc}^*$ (as $\mathbf{SupQ}^*\mathbf{Bc}^* \subseteq \mathbf{TxtBc}^*$, see Theorem 57) or against $\mathbf{ResSupQ}^*\mathbf{Ex}$ (as $\mathbf{LSupQ}^*\mathbf{I} = \mathbf{SupQ}^*\mathbf{I} = \mathbf{ResSupQ}^*\mathbf{I}$, see Proposition 41).

Theorem 37. *For all $n \in N$, there exists a $\mathcal{L}$ such that*

(a) for all $t \in N$, $\mathcal{L} \notin \mathbf{ResSupQ}^n\mathbf{Bc}^t$;

(b) $\mathcal{L} \notin \mathbf{ResSupQ}^n\mathbf{Ex}^*$;

(c) $\mathcal{L} \in \mathbf{SupQ}^2\mathbf{Ex}$.

Proof. Let

$\mathcal{C} = \{L \mid$

Let $S = \{i \mid L \cap \mathrm{CYL}_i \neq \emptyset\}$.

Let $e = \max(S)$.

1. $\mathrm{card}(S) = n + 1$.
2. $(L - \mathrm{CYL}_e)$ is finite.
3. Either

3.1 W_e is infinite and $L \cap \mathrm{CYL}_e = \{\langle e, x\rangle \mid x \in W_e\}$.

or

3.2 W_e is finite, and $(\exists w)[$
$L \cap \{\langle e, 2w\rangle, \langle e, 2w+1\rangle\} = \emptyset$ and
$(\forall x < w)[L \cap \{\langle e, 2x\rangle, \langle e, 2x+1\rangle\} \neq \emptyset]$, and
$(\forall y > w)[\langle e, 2y\rangle \in L \ \wedge \ \langle e, 2y+1\rangle \notin L]]$.

]}

$\mathcal{C}$ can be shown to be in $\mathbf{SupQ^1Ex}$, using the same methods as in Theorem 22. $\mathcal{C} \notin \bigcup_{t\in N} \mathbf{TxtBc}^t \cup \mathbf{TxtEx}^*$, can be proved along the same lines as Lemma 20 and Corollary 21. Further note that $\mathcal{C}$ has the property that, for all finite sets B of size at most n, $\mathcal{C} \cap \{L \mid B \subseteq L\} \notin \bigcup_{t\in N} \mathbf{TxtBc}^t \cup \mathbf{TxtEx}^*$.

It is easy to verify that, for any fixed $r \in N$, $\mathcal{C}_r = \{L' \mid (\exists L \in \mathcal{C})[L' = \{\langle r, x\rangle \mid x \in L\}]\}$ is also in $\mathbf{SupQ^1Ex}$, and such a learner can be found effectively from r and a $\mathbf{SupQ^1Ex}$-learner for $\mathcal{C}$. Similarly, it can be shown that $\mathcal{C}_r \notin \bigcup_{t\in N} \mathbf{TxtBc}^t \cup \mathbf{TxtEx}^*$ (where we additionally have that for any $B \subseteq \mathrm{CYL}_r$ of size at most n, $\mathcal{C}_r \cap \{L \mid B \subseteq L\} \notin \bigcup_{t\in N} \mathbf{TxtBc}^t \cup \mathbf{TxtEx}^*$).

Let $A = \bigcup_{j\in K} \mathrm{CYL}_j$.

$\mathcal{L} = \{L \cup A \mid (\exists r \notin K)[L \in \mathcal{C}_r]\}$.

Claim 38. $\mathcal{L} \in \mathbf{SupQ}^2\mathbf{Ex}$.

Proof. A $\mathbf{SupQ}^2\mathbf{Ex}$ learner $\mathbf{M}'$ for $\mathcal{L}$ can be constructed as follows. $\mathbf{M}'$ first asks a question whether A is superset of the input language. Suppose $\langle r, x\rangle$ is the counterexample (there must be such a counterexample for input languages from $\mathcal{L}$). For a segment σ, let σ' be obtained by converting any $\langle r', x\rangle$, $r \neq r'$ into #. Let f be a recursive function such that $W_{f(j)} = W_j \cup A$. Let $\mathbf{M}$ be a $\mathbf{SupQ^1Ex}$-learner for $\mathcal{C}_r$ (note that such a learner can be effectively found from r). Now on input σ, $\mathbf{M}'(\sigma)$ simulates $\mathbf{M}(\sigma')$. If $\mathbf{M}$ asks a question for W_j, then $\mathbf{M}'$ asks a question for $W_{f(j)}$, and passes the answer it receives to $\mathbf{M}$. If $\mathbf{M}$ conjectures j as a grammar, then $\mathbf{M}'$ conjectures $f(j)$. It is then easy to verify that $\mathbf{M}'$ $\mathbf{SupQ}^2\mathbf{Ex}$-identifies $\mathcal{L}$.

Claim 39. *(a) For all* $t \in N$, $\mathcal{L} \notin \mathbf{ResSupQ}^n\mathbf{Bc}^t$.

(b) $\mathcal{L} \notin \mathbf{ResSupQ}^n\mathbf{Ex}^*$.

Proof. We only show part (a). Part (b) can be shown similarly.

Suppose by way of contradiction $\mathbf{M}$ $\mathbf{ResSupQ}^n\mathbf{Bc}^t$-identifies $\mathcal{L}$. Without loss of generality assume that $\mathbf{M}'$ never asks more than n questions, irrespective of whether the input is from $\mathcal{L}$ of not, or if the answers are wrong, or even if the answers are inconsistent.

Subclaim: We first claim that there must be a τ, $\mathrm{content}(\tau) \subseteq A$, such that $\mathbf{M}$ does not ask questions on any extension τ' of τ with $\mathrm{content}(\tau') \subseteq A$, whatever answers one may have given to $\mathbf{M}$ on the earlier questions on prefixes of τ. To see this, consider a tree T_σ formed as follows for any σ (we only care about T_σ, for $\mathrm{content}(\sigma) \subseteq A$). Nodes of the tree have labels of the form (σ', a finite string over $\{Yes, No\}$), where $\sigma' \subseteq \sigma$. The node $(\sigma', s = s_1s_2 \ldots s_k)$ signifies the following: If s is empty, then σ' must be the smallest prefix of σ on which $\mathbf{M}$ asked a question (also (σ', Λ) must be the root of T_σ). Children (if any) of a node $(\sigma', s = s_1s_2 \ldots s_k)$ are of form $(\sigma'', s = s_1s_2 \ldots s_ks_{k+1})$, where $\sigma' \subseteq \sigma'' \subseteq \sigma$, and if the questions of $\mathbf{M}$ on prefixes of σ' (which are from the first component of the nodes on the path from root to $(\sigma', s = s_1s_2 \ldots s_k)$) are answered as $s_1s_2 \ldots s_ks_{k+1}$, (i.e., first question is answered s_1, second question is answered s_2, $\cdots$, the $(k+1)$-th (which is at σ') is answered s_{k+1}), then $\mathbf{M}$ asks a question at σ'', but not at any σ''', with $\sigma' \subset \sigma''' \subset \sigma''$. Intuitively, T_σ just shows the tree of questions asked on prefixes of σ, where answers may be given in all possible ways. It is easy to verify that $T_\sigma \subseteq T_\gamma$, for $\sigma \subseteq \gamma$. Moreover, there exists a maximal tree as none of T_σ can have more than 2^n nodes, due to the bound on the number of questions.

Now any τ such that—(i) $\mathrm{content}(\tau) \subseteq A$ and (ii) for all $\tau' \supseteq \tau$ with $\mathrm{content}(\tau') \subseteq A$, $T_\tau = T_{\tau'}$—satisfies the requirements of the subclaim.

We now continue with the proof of the claim. Choose τ as guaranteed by the subclaim above. Now let $S = \{r \mid (\exists \tau' \supset \tau)[\text{content}(\tau') \subseteq A \cup \text{CYL}_r, \mathbf{M}(\tau') \text{ asks a question, for some way of answering questions on prefixes of } \tau]\}$. As $S \subseteq \overline{K}$, and S is r.e., we must have an r such that $r \in \overline{K} - S$.

Thus, $\mathbf{M}$ does not ask any further question on any text T extending τ, for languages L satisfying $\text{content}(\tau) \subseteq L \subseteq A \cup \text{CYL}_r$.

Now consider answering the questions on prefixes of τ as follows: If the query Q contains $A \cup \text{CYL}_r$, then answer yes. Otherwise answer no. Let X consist of least elements in $(A \cup \text{CYL}_r) - Q$, for each query Q answered no above. Note that X contains at most n elements.

Now $\mathbf{M}$ has to $\mathbf{Bc}^t$-identify any member L of $\mathcal{L}$ satisfying $A \cup X \subseteq L \subseteq A \cup \text{CYL}_r$, without asking any further queries. Thus, $\mathcal{L} \cap \{L \mid A \cup X \subseteq L \subseteq A \cup \text{CYL}_r\} \in \mathbf{TxtBc}^t$, and hence $\mathcal{C}_r \cap \{L \mid X \cap \text{CYL}_r \subseteq L\} \in \mathbf{TxtBc}^t$. However this is not possible, by definition of $\mathcal{C}$ and $\mathcal{C}_r$.

This completes the proof of the claim and the theorem. □

7. Learning via subset queries versus learning with bounded number of negative counterexamples to conjectures

We first prove some useful propositions.

Proposition 40. *For any* $a \in N \cup \{*\}, \mathbf{I} \in \{\mathbf{Ex}^a, \mathbf{Bc}^a\}$,
(a) $\mathbf{SubQ}^*\mathbf{I} \subseteq \mathbf{NCI}$.
(b) $\mathbf{LSubQ}^*\mathbf{I} \subseteq \mathbf{LNCI}$.
(c) $\mathbf{ResSubQ}^*\mathbf{I} \subseteq \mathbf{ResNCI}$.

Proof. We show part (a). Parts (b) and (c) can be shown similarly. An $\mathbf{NCI}$ learner could just conjecture the query of the $\mathbf{SubQ}^*\mathbf{I}$ learner to obtain negative counterexamples, if any for the queries of $\mathbf{SubQ}^*\mathbf{I}$ learner. Thus, the proposition holds. □

Proposition 41. *Suppose* $a \in N \cup \{*\}, \mathbf{I} \in \{\mathbf{Ex}^a, \mathbf{Bc}^a\}$.
(a) $\mathbf{ResSubQ}^*\mathbf{I} = \mathbf{SubQ}^*\mathbf{I} = \mathbf{LSubQ}^*\mathbf{I}$.
(b) $\mathbf{ResSupQ}^*\mathbf{I} = \mathbf{SupQ}^*\mathbf{I} = \mathbf{LSupQ}^*\mathbf{I}$.

Proof. (a) Clearly, $\mathbf{ResSubQ}^*\mathbf{I} \subseteq \mathbf{SubQ}^*\mathbf{I} \subseteq \mathbf{LSubQ}^*\mathbf{I}$.

To show that $\mathbf{LSubQ}^*\mathbf{I} \subseteq \mathbf{ResSubQ}^*\mathbf{I}$, note that for any W_i, if $W_i \not\subseteq \text{content}(T)$, then one can find in the limit, from a text T, the least element in $W_i - \text{content}(T)$. Thus, one can eventually answer correctly all the queries of a $\mathbf{LSubQ}^*\mathbf{I}$-learner, using a $\mathbf{ResSubQ}^*\mathbf{I}$-learner. Part (a) follows.

(b) can be proved similarly. □

Proposition 42. *Suppose* $a \in N \cup \{*\}$. $\mathbf{NCEx}^a = \mathbf{SubQ}^*\mathbf{Ex}^a = \mathbf{LNCEx}^a = \mathbf{LSubQ}^*\mathbf{Ex}^a = \mathbf{ResNCEx}^a = \mathbf{ResSubQ}^*\mathbf{Ex}^a$.

Proof. By Propositions 40 and 41, it is enough to show $\mathbf{LNCEx}^a \subseteq \mathbf{LSubQ}^*\mathbf{Ex}^a$.

Suppose $\mathcal{L} \in \mathbf{LNCEx}^a$ as witnessed by machine $\mathbf{M}$. An $\mathbf{LSubQ}^*\mathbf{Ex}^a$-learner can provide the counterexamples to $\mathbf{M}$ by just asking subset query for each of the conjectures of $\mathbf{M}$. Proposition follows. □

Our main result in this section shows that, surprisingly, there is a class of languages that can be **Ex**-learned using just one subset (or equivalence) restricted query, but cannot be **Bc***-learned using any bounded number of negative counterexamples to conjectures! Intuitively, the teacher helping to learn this class of languages, being asked subset query for every conjecture, is forced to output n negative counterexamples, while just one "wise" subset query might be enough.

Theorem 43. *For all* $n \in N$, $(\mathbf{ResSubQ}^1\mathbf{Ex} \cap \mathbf{ResEquQ}^1\mathbf{Ex}\ ) - \mathbf{LNC}^n\mathbf{Bc}^* \neq \emptyset$.

Proof. For $A \subseteq N$,

Let $X_A = \{\mathrm{CYL}_i \mid i \in A\}$.

Let $Y_A^F = X_A - F$.

Let $Z_A^{F,B} = (X_A - F) \cup B$.

$\mathcal{L} = \{Y_A^F \mid \mathrm{card}(A) \leqslant n+1, F = D_{\max(A)}\} \cup \{Z_A^{F,B} \mid (\exists i)[\mathrm{card}(A) = n, \max(A) < i, F = D_i, \text{ and } B \text{ is a non-empty finite subset of } \mathrm{CYL}_i]\}$.

Intuitively, for $L \in \mathcal{L}$, either L consists of upto $n+1$ cylinders, with some elements missing, or it consists of n cylinders, with some elements missing, plus a finite portion of another cylinder. The missing elements mentioned above are coded using the maximum index cylinder present in L. This allows for easy learnability, as long as one can determine whether the $(n+1)$-th cylinder, if any, is present fully in the input, or only finite portion of it is in the input. This can be done using subset or equivalence query. On the other hand, the missing elements can force a **LNC**-type learner to make enough (n) non-subset conjectures, and thus not able to determine, whether the $(n+1)$-th cylinder is present in full or only partially. This allows for diagonalization. We now proceed formally.

Claim 44. $\mathcal{L} \in \mathbf{ResSubQ}^1\mathbf{Ex} \cap \mathbf{ResEquQ}^1\mathbf{Ex}$.

Proof. On input σ, the learner behaves as follows.

Let $A = \{i \mid \mathrm{CYL}_i \cap \mathrm{content}(\sigma) \neq \emptyset\}$. If $\mathrm{card}(A) \leqslant n$, then output a grammar for $Y_A^{D_{\max(A)}}$.

If $\mathrm{card}(A) = n+1$, then let $i = \max(A)$. The learner asks a (subset/equivalence) query (assuming no previous query) about $Y_A^{D_i}$. If answer is yes, then the learner continues outputting a grammar for $Y_A^{D_i}$. If no, then learner outputs a grammar for $Y_{A-\{i\}}^{D_i} \cup (\mathrm{content}(\sigma) \cap \mathrm{CYL}_i)$. It is easy to verify that the learner $\mathbf{SubQ}^1\mathbf{Ex}$-identifies ($\mathbf{EquQ}^1\mathbf{Ex}$-identifies) $\mathcal{L}$.

Claim 45. $\mathcal{L} \notin \mathbf{LNC}^n\mathbf{Bc}^*$.

Proof. Suppose by way of contradiction that $\mathbf{M}$ witnesses that $\mathcal{L} \in \mathbf{LNC}^n\mathbf{Bc}^*$. Let $\sigma_0 = \sigma_0' = \Lambda$. Let $F_0 = \{\langle 0,0\rangle\}$. Let $R_0 = \{i\}$, where $D_i = F_0$.

Inductively define $\sigma_{i+1}, R_{i+1}, F_{i+1}$, for $i < n$ as follows.

(* The construction is non-effective. *)

(* The following invariants will be satisfied:

(a) $\mathrm{content}(\sigma_i) \subseteq Y_{R_i - \{\max(R_i)\}}^{F_i}$.

(b) $F_i = D_{\max(R_i)}$.

(c) $\mathrm{content}(\sigma_{i'})$ contains i elements.

(d) The counterexample sequence $\sigma_{i'}$ is consistent with the input language being any L such that $\mathrm{content}(\sigma_i) \subseteq L \subseteq N - F_i$.

*)

1. Check whether there exists a $\sigma \supseteq \sigma_i$ such that $\text{content}(\sigma) \subseteq Y_{R_i}^{F_i}$, and $W_{\mathbf{M}(\sigma,\sigma_i'\#^{|\sigma|-|\sigma_i|})} \not\subseteq \text{content}(\sigma)$.
   If there is no such σ, then $\sigma_{i'}$, $i' > i$ do not get defined.
   If there exists such a σ, then choose a shortest such σ, and proceed as follows.
2. Let $\langle j,k\rangle$ be the least element of $W_{\mathbf{M}(\sigma,\sigma_i'\#^{|\sigma|-|\sigma_i|})} - \text{content}(\sigma)$.
   Let $\sigma_{i+1} = \sigma\#$ and $\sigma_{i+1}' = \sigma_i'\#^{|\sigma|-|\sigma_i|}\langle j,k\rangle$. (* That is, we give $\langle j,k\rangle$ as counterexample *).
   Pick $z \notin \text{content}(\sigma)$ such that
   for $D_r = F_i \cup \{\langle j,k\rangle\} \cup \{z\}$, $r > \max(R_i \cup D_r)$.
   (* Note that there clearly exist such z, r. *)
   Let $F_{i+1} = D_r$ and $R_{i+1} = R_i \cup \{r\}$.

End

It is easy to verify that the invariants are satisfied.

Now, if σ_n gets defined, then clearly $\mathbf{M}$ has already received n negative counterexamples. Thus, $\mathbf{M}$ now needs to $\mathbf{Bc}^*$-identify (without receiving any more negative examples) the languages $Y_{R_n}^{F_n}$, and $Z_{R_n-\{\max(R_n)\}}^{F_n,B}$, such that B is a non-empty finite subset of $\text{CYL}_{\max(R_n)}$. This is not possible by Proposition 11.

On the other hand, if σ_n does not get defined, then let m be the largest number such that σ_m gets defined. Now, due to non-success of the search for σ (for the definition of σ_{m+1}), we have that $\mathbf{M}$ does not $\mathbf{LNC}^n\mathbf{Bc}^*$-identify $Y_{R_m}^{F_m}$. □

Now we show that one negative answer to conjecture (or one restricted subset or equivalence query) in the context of **Ex**-learning can sometimes do more than any number of counterexamples (to conjectures) of bounded size, even in the context of $\mathbf{Bc}^*$-learning.

Theorem 46. $(\mathbf{ResSubQ}^1\mathbf{Ex} \cap \mathbf{ResEquQ}^1\mathbf{Ex} \cap \mathbf{ResNC}^1\mathbf{Ex}) - \mathbf{BNCBc}^* \neq \emptyset$.

Proof. The class $\mathcal{L} = \{N\} \cup \mathbf{FINITE}$ is clearly in $\mathbf{ResEquQ}^1\mathbf{Ex} \cap \mathbf{ResNC}^1\mathbf{Ex} \cap \mathbf{ResSubQ}^1\mathbf{Ex}$. $\mathcal{L} \notin \mathbf{BNCBc}^*$, was shown in [19]. □

In contrast to our Theorem 43, the following result shows that sometimes just one counterexample to conjecture can do more than any number of subset queries receiving the least counterexamples.

Theorem 47. *For all* $n \in N$, $\mathbf{ResNC}^1\mathbf{Ex} - \mathbf{LSubQ}^n\mathbf{Bc}^* \neq \emptyset$.

Proof. For $A \subseteq N$, and any set B, let $X_A = \bigcup_{i\in A} \text{CYL}_i$, and $Y_{A,B} = X_A \cup B$.

$\mathcal{L} = \{X_A \mid \{0\} \subseteq A \subseteq N, \text{card}(A) < \infty\} \cup \{Y_{A,B} \mid A, B \text{ are finite, } \{0\} \subseteq A, \text{ and } (\exists i > \max(A))[B \subseteq \text{CYL}_i]\}$.

Intuitively, the languages L in $\mathcal{L}$ consist of elements from finitely many cylinders, all of which (except maybe one) are fully in the language L. Furthermore, if a cylinder is only partially in L, then it must be the one with largest index, and only finitely many elements from it are in L. This allows for easy learning using one counterexample in **ResNCEx** model. However, for suitably chosen diagonalizing language L, a $\mathbf{LSubQ}^n\mathbf{Bc}^*$ learner cannot obtain relevant information to distinguish whether the highest indexed cylinder is fully or partially in the input language. This allows us to show that $\mathcal{L} \notin \mathbf{LSubQ}^n\mathbf{Bc}^*$.

Claim 48. $\mathcal{L} \in \mathbf{ResNC}^1\mathbf{Ex}$.

Proof. We first show that $\mathcal{L} \in \mathbf{BNC}^1\mathbf{Ex}$. On input (σ, σ'), compute $A = \{i \mid \langle i, x\rangle \in \text{content}(\sigma)\}$. If $\text{content}(\sigma') = \emptyset$, then conjecture a grammar for X_A. If $\text{content}(\sigma') \neq \emptyset$, then conjecture a grammar for $Y_{A-\{\max(A)\},B}$, where $B = \text{content}(\sigma) \cap \text{CYL}_{\max(A)}$.

It is now easy to verify that $\mathbf{M}$ gets at most one counterexample, and identifies the class $\mathcal{L}$ (if input language is not of form X_A, then it will eventually get a counterexample, as all languages in $\mathcal{L}$ are infinite). As the above construction does not use the exact value of the counterexample, it follows that $\mathcal{L} \in \mathbf{ResNC}^1\mathbf{Ex}$ also.

Claim 49. $\mathcal{L} \notin \mathbf{LSubQ}^n\mathbf{Bc}^*$.

Proof. Suppose by way of contradiction $\mathbf{M}$ learns $\mathcal{L}$ using at most n subset queries.

We will use variables R_i, S_i below. Intuitively, for the constructed diagonalizing language, for A as in definition of $\mathcal{L}$, we would have $R_i - \{\max(R_i)\} \subseteq A$, and $S_i \cap A = \emptyset$. Let $\sigma_0 = \Lambda$. Let $S_0 = \emptyset$ and $R_0 = \{0, 1\}$.

Inductively define σ_{i+1} (along with R_{i+1}, S_{i+1}), for $i < n$, as follows.
(* The construction is non-effective. *)
(* Following invariants will be satisfied:
(a) $R_i \cap S_i = \emptyset$.
(b) $\text{content}(\sigma_i) \subseteq X_{R_i-\{\max(R_i)\}}$.
(c) $\mathbf{M}$ has already made i queries on proper prefixes of σ_i.
(d) Answers given to $\mathbf{M}$ are consistent with input being any language L such that $X_{R_i-\{\max(R_i)\}} \subseteq L \subseteq N - X_{S_i}$.
*)
1. Check if there exists a $\sigma \supseteq \sigma_i$ such that $\text{content}(\sigma) \subseteq X_{R_i}$, and $\mathbf{M}$ makes a subset query on σ.
If there is no such σ, then $\sigma_{i'}$, $i' > i$ do not get defined.
If there exists such a σ, then choose a shortest such σ, and proceed as follows.
2. Let Q be the queried language.
3.1 If $Q - X_{R_i} \neq \emptyset$, Then
Answer the query no, with $\langle j, x\rangle = \min(Q - X_{R_i})$ as a counterexample.
Let $\sigma_{i+1} = \sigma\#$,
Let $S_{i+1} = S_i \cup \{j\}$, and $R_{i+1} = R_i \cup \{j'\}$, where $j' > \max(R_i \cup S_{i+1})$.
3.2 Else (i.e., $Q \subseteq X_{R_i}$)
Answer the query yes.
Let $\sigma_{i+1} = \sigma\#$,
Let $S_{i+1} = S_i$ and $R_{i+1} = R_i \cup \{j'\}$ such that $j' > \max(R_i \cup S_i)$,
End

It is easy to verify that invariants are satisfied. Let m be largest value such that σ_m is defined. Note that $\mathbf{M}$ does not ask any queries on $\sigma \supseteq \sigma_m$, such that $\text{content}(\sigma) \subseteq X_{R_m}$ (if $m = n$, due to bound on number of queries, $\mathbf{M}$ cannot make any more queries; if $m < n$, the failure of search for $\sigma \supseteq \sigma_m$, in which $\mathbf{M}$ asks a query, implies that $\mathbf{M}$ does not make any more queries).

Now $\mathbf{M}$ needs to $\mathbf{Bc}^*$-identify, without any more queries, X_{R_m}, as well as $Y_{R_m-\{\max(R_m)\},B}$ for all finite $B \subseteq \text{CYL}_{\max(R_m)}$. This is not possible by Proposition 11. □

Above proof also shows $\mathbf{BNC}^1\mathbf{Ex} - \mathbf{LSubQ}^n\mathbf{Bc}^* \neq \emptyset$.

Note that the above theorem cannot be generalized to provide $\mathbf{ResNC^1Ex} - \mathbf{LSubQ^*Bc^*} \neq \emptyset$, as $\mathbf{LSubQ^*Ex}^a = \mathbf{NCEx}^a$, by Proposition 42. However, we can do the diagonalization against $*$-number of subset queries, if we consider **NCBc** model.

Theorem 50. $\mathbf{NC^1Bc} - \mathbf{SubQ^*Bc^*} \neq \emptyset$.

Proof. Consider the following class of languages.

$\mathcal{L}_1 = \{L \mid L \cap \mathrm{CYL}_0$ is infinite, and for all p such that $\langle 0, p\rangle \in L$, $W_p = L\}$.

$\mathcal{L}_2 = \{L \mid L \cap \mathrm{CYL}_0$ is non-empty and finite, and for all p such that $\langle 0, p\rangle \in L$, $W_p \subseteq L$ and for $p = \max(\{x \mid \langle 0, x\rangle \in L\})$, $W_p = L\}$.

$\mathcal{L}_3 = \{L \mid L = \mathrm{CYL}_1 \cup C$ for some finite C, and $\{x \mid \langle 0, x\rangle \in L\} \neq \emptyset$ and for $p = \max(\{x \mid \langle 0, x\rangle \in L\})$, $W_p \not\subseteq L]\}$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2 \cup \mathcal{L}_3$.

Claim 51. $\mathcal{L} \in \mathbf{NC^1Bc}$.

Proof. Consider a learner which outputs on input (σ, σ') as follows. If $\mathrm{content}(\sigma') \neq \emptyset$, then output a grammar for $\mathrm{CYL}_1 \cup \mathrm{content}(\sigma)$. Otherwise, if $\mathrm{content}(\sigma) \cap \mathrm{CYL}_0 = \emptyset$, then output a grammar for $\emptyset$; else output p, where $p = \max(\{x \mid \langle 0, x\rangle \in \mathrm{content}(\sigma)\})$.

Now consider any text T for a language $L \in \mathcal{L}$. Clearly, if the input language is in $\mathcal{L}_1$ or $\mathcal{L}_2$, then there will never be a counterexample, and the learner would output a correct grammar for L, on input $T[n]$, for all but finitely many n.

On the other hand, if $L \in \mathcal{L}_3$, then there will be a counterexample eventually given to the above learner, and thus, for all but finitely many n, on input $T[n]$, the learner will output a grammar for $\mathrm{CYL}_1 \cup \mathrm{content}(T[n])$. It follows that the learner $\mathbf{NC^1Bc}$ -identifies $\mathcal{L}$.

Claim 52. $\mathcal{L} \notin \mathbf{SubQ^*Bc^*}$.

Proof. Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{SubQ^*Bc^*}$-identifies $\mathcal{L}$. Then by Operator recursion theorem [9], there exists a 1–1 increasing function p such that $W_{p(i)}$ may be defined as follows.

Initially, let σ_0 and $W_{p(0)}$ contain just $\langle 0, p(0)\rangle$. Let $W^s_{p(i)}$ denote $W_{p(i)}$ enumerated before stage s. Initially let $B_0 = \emptyset$. Intuitively, B_s denotes the set of elements we have decided to keep out of the diagonalizing language.

For all j, initially let $F(j) = \#$. Intuitively, $F(j)$ denotes the answer to subset query W_j. Initially all the queries are answered as yes. During the construction, some of these answers may be changed to no, and $F(j)$ updated to a negative counterexample for W_j. It will be the case that value of $F(j)$, once changed to a number from N, will never change again. Go to stage 0.

Stage s

(* Following invariants will be maintained by the construction:

(a) $W^s_{p(0)} = \mathrm{content}(\sigma_s)$ and $\mathrm{content}(\sigma_s) \cap B_s = \emptyset$.

(b) For all j, $F(j)$ is either # or a member of B_s. In case $F(j) \in B_s$, then W_j contains $F(j)$.

(c) For any element $w \in B_s$, we have $\langle 0, p(s')\rangle > w$ and $\langle 2, s'\rangle > w$, for all $s' > s$.

(d) If $\langle 0, x\rangle \in \mathrm{content}(\sigma_s)$, then $x = p(0)$ or $x = p(i)$, for some $i \leqslant s$, and $W_{p(i)}$ was made equal to $W_{p(0)}$ in stage $i - 1$, step 5.

*)

1. Let $W_{p(s+1)}$ enumerate content$(\sigma_s) \cup \langle 0, p(s+1)\rangle$. Dovetail steps 2, 3 and 4 until either step 3 or step 4 succeeds. If step 3 succeeds before step 4 (if ever) then go to step 5. If step 4 succeeds before step 3 (if ever) then go to step 6. Here we assume that if the search in step 4 can succeed within s steps, then it succeeds before step 3 (thus some priority is given to the search in step 4).
   (* Below, for simulating **M** on any input τ, we assume that answers are given according to F for the queries made by **M** on prefixes of τ. That is, if $F(j) = \#$, then answer is yes, and if $F(j) \in N$, then answer is no, with $F(j)$ being the counterexample. *)
2. For $t = 0$ to ∞ do
   Enumerate $\langle 1, t\rangle, \langle 2, t+1+s\rangle$ in $W_{p(s+1)}$.
   Endfor
3. Search for a $\sigma \supseteq \sigma_s$, such that content$(\sigma) \subseteq$ (content$(\sigma_s) \cup \mathrm{CYL}_1 \cup \mathrm{CYL}_2 \cup \{\langle 0, p(s+1)\rangle\}) - B_s$ such that, $\mathbf{M}(\sigma)$ asks a question on σ.
4. Search for a query W_j made on some prefix of σ_s, such that $F(j) = \#$, but W_j enumerates an element $x \notin$ content$(\sigma_s) \cup \mathrm{CYL}_1$.
   (* Here we assume without loss of generality that, if there exists a query $W_{j'}$ made by **M** on a prefix of σ_s such that $F(j') = \#$, and $W_{j'} - (\text{content}(\sigma_s) \cup \mathrm{CYL}_1) \neq \emptyset$, then the above search will succeed with $j = j'$ for earliest such query made (i.e., query $W_{j'}$ made on shortest prefix of σ_s). *)
5. Enumerate elements of content(σ) and $W_{p(s+1)}$ enumerated until now into $W_{p(0)}$. From now on $W_{p(s+1)}$ enumerates whatever $W_{p(0)}$ enumerates.
   (* Thus, $W_{p(s+1)} = W_{p(0)}$. *)
   Let $B_{s+1} = B_s$.
   Let σ_{s+1} be an extension of σ such that content$(\sigma_{s+1}) = W_{p(0)}$ enumerated upto now.
   Go to stage $s+1$.
6. Let W_j be as found in step 4.
   Let w be such that $w \in W_j - (\text{content}(\sigma_s) \cup \mathrm{CYL}_1)$. Change $F(j)$ to w.
   (* Note that this changing of answer would change the behaviour of **M** on later part σ_s. *)
   (* Following is done to avoid enumerating w in $W_{p(0)}$ in any future stages, and maintain invariant (c). *)
   Let $s' > s$ be such that $\langle 0, p(s')\rangle$ and $\langle 2, s'\rangle$ are both $> w$.
   Let $B_{s'} = B_s \cup \{w\}$.
   Let $\sigma_{s'} = \sigma_s$.
   Go to stage s' (i.e., we assume that the stages $s < s'' < s'$, are just null stages, with corresponding $B_{s'}, \sigma_{s'}$ being just B_s, σ_s).

End stage s

It is easy to verify that invariants are maintained by the construction.

We now consider the following cases.

Case 1: Some stage s starts but does not finish.

In this case consider any language L which satisfies:

$L = W_{p(s+1)}$ (which is in $\mathcal{L}_2$) or

$L =$ content$(\sigma_s) \cup \mathrm{CYL}_1 \cup \{\langle 0, p(s+1)\rangle\} \cup C$, for some finite $C \subseteq W_{p(s+1)}$ (which are in $\mathcal{L}_3$).

Now, for each such L, on any text T for L which extends σ_s, $\mathbf{M}$ does not ask any further queries on T beyond σ_s. Also the answer given to $\mathbf{M}$ on the queries made on prefixes of σ_s are correct as step 4 did not succeed. It follows that $\mathbf{M}$ needs to $\mathbf{Bc}^*$ identify all such L without asking any further queries. However this is not possible by Proposition 11.

Case 2: There exist infinitely many stages.

In this case let $L = W_{p(0)}$. It is easy to verify that $L \in \mathcal{L}_1$, as for every $\langle 0, p(s+1)\rangle$, enumerated in $W_{p(0)}$ we have $W_{p(s+1)} = W_{p(0)}$, due to $W_{p(s+1)}$ eventually following $W_{p(0)}$ by step 5 in stage s.

Let $T = \bigcup_{s\in N} \sigma_s$.

Note that due to priority given to step 4 above, we can show by induction that for all i, eventually, the first i questions $W_{j_1}, W_{j_2}, \ldots, W_{j_i}$ asked by $\mathbf{M}$ on T would be answered correctly using F. It follows that all questions asked by $\mathbf{M}$ on T are eventually answered correctly using F.

Let $F'(j)$ denote the final value of $F(j)$ as given by the above construction. We now claim that $\mathbf{M}$ would ask infinitely many questions on T when answers are given using F'. To see this, suppose by way of contradiction otherwise. Let s be large enough so that $\mathbf{M}$ will not ask any questions beyond σ_s, if answers are given according to F'. Let $s' > s$ be large enough so that all the answers given according to F in stage s' would be correct for questions asked by $\mathbf{M}$ on σ_s. But then the steps 3 and 4 in the construction would not succeed in stage s', contradicting the hypothesis of having infinitely many stages.

Claim follows from the above two cases. □

Corollary 53. $\mathbf{NC}^1\mathbf{Bc} - \mathbf{LSubQ}^*\mathbf{Bc}^* \neq \emptyset$.

Proof. Follows using Theorem 50 and Proposition 41. □

Proof of Theorem 50 also shows $\mathbf{BNC}^1\mathbf{Bc} - \mathbf{LSubQ}^*\mathbf{Bc}^* \neq \emptyset$.

8. Subset queries and NC versus other types of queries

We first consider diagonalization against equivalence queries. As $\mathbf{ResEquQ}^*\mathbf{Ex}$ contains the class $\mathcal{E}$, this diagonalization can only be done against bounded number of equivalence queries.

Proposition 54. $\mathcal{E} \in \mathbf{ResEquQ}^*\mathbf{Ex}$.

Proof. A learner can $\mathbf{ResEquQ}^*\mathbf{Ex}$-learn all the r.e.languages, by sequentially asking equivalence queries for $W_0, W_1, W_2, \ldots$, until an i is found such that $W_i =$ input language. When such a i is found, the learner conjectures grammar i from then onwards. □

The following proposition is useful to prove Theorem 56.

Proposition 55. *Suppose $n \in N$, A is an infinite- coinfinite language, $B \subseteq A$, and $A - B$ is infinite. Then, $\mathcal{L} = \{B \cup C \mid \mathrm{card}(C) < \infty\} \cup \{A \cup C \mid \mathrm{card}(C) < \infty\} \notin \mathbf{LEquQ}^n\mathbf{Bc}^*$.*

Proof. Suppose by way of contradiction that $\mathbf{M}$ witnesses that $\mathcal{L} \in \mathbf{LEquQ}^n\mathbf{Bc}^*$.

We will use variables R_i and S_i below. Intuitively, we have committed R_i to be in the diagonalizing language and S_i to be out of the diagonalizing language being constructed. Initially let $\sigma_0 = \Lambda$, and $R_0 = \emptyset, S_0 = \emptyset$.

Inductively define σ_{i+1}, for $i < n$, as follows.
(* The construction is non-effective. *)
(* We will maintain the following invariants:
(a) $\text{content}(\sigma_i) \subseteq R_i$.
(b) $(A \cup R_i) \cap S_i = \emptyset$.
(c) $\mathbf{M}$ has already asked i queries on proper prefixes of σ_i.
(d) Answers given to queries of $\mathbf{M}$ are consistent with any input language L satisfying: $R_i \subseteq L \subseteq N - S_i$.
*)
1. Check if there exists a $\sigma \supseteq \sigma_i$, such that $\text{content}(\sigma) \cap S_i = \emptyset$, and $\mathbf{M}$ asks a query on input σ.
If there is no such σ, then $\sigma_{i'}$, $i' > i$ do not get defined.
If there exists such a σ, then choose a shortest such σ and proceed as follows.
2. Let Q be the queried language.
3. Answer the query as no.
Pick least w such that one of the following holds:
i) $w \in \text{content}(\sigma) \cup R_i \cup A$ and $w \notin Q$.
ii) $w \in S_i$ and $w \in Q$,
iii) $w \notin \text{content}(\sigma) \cup R_i \cup S_i \cup A$.
4. Give this w as counterexample to $\mathbf{M}$ for the query Q.
5. Let $\sigma_{i+1} = \sigma\#$.
(* Note that we need σ_{i+1} to properly extend σ, for search at step 1 of next iteration. *)
6. If $w \in Q$, then let
$S_{i+1} = S_i \cup \{w\}$ and
$R_{i+1} = R_i \cup \text{content}(\sigma) \cup \{w' < w \mid w' \in A\}$.
(* $\{w' < w \mid w' \in A\}$ is added to diagonalizing language to make sure that w is indeed the least counterexample to query Q by $\mathbf{M}$. *)
If $w \notin Q$, then let
$S_{i+1} = S_i$ and
$R_{i+1} = R_i \cup \{w\} \cup \text{content}(\sigma) \cup \{w' < w \mid w' \in A\}$.

End

It is easy to verify that the above construction maintains the invariants.

Let m be the largest number such that σ_m gets defined. Now $\mathbf{M}$ does not ask any more questions on $\sigma \supseteq \sigma_m$ such that $\text{content}(\sigma) \subseteq N - S_m$ (if $m = n$, then $\mathbf{M}$ has already asked n questions; if $m < n$, then due to non-success in search for σ during the definition of σ_{i+1}, we have that $\mathbf{M}$ does not ask any more questions).

Thus, $\mathbf{M}$ must $\mathbf{Bc}^*$-identify (without any further queries) the class $\{A \cup R_m\} \cup \{B \cup R_m \cup C \mid C \subseteq A, \text{card}(C) < \infty\}$, an impossible task by Proposition 11. □

The following theorem demonstrates that sometimes $\mathbf{Ex}$-learners using just one restricted subset query or getting just one bounded negative counterexample to conjectures can do better than any $\mathbf{Bc}^*$-learner, asking at most n equivalence queries and receiving least counterexamples.

Theorem 56. $\mathbf{ResSubQ}^1\mathbf{Ex} \cap \mathbf{ResNC}^1\mathbf{Ex} - \mathbf{LEquQ}^n\mathbf{Bc}^* \neq \emptyset$.

Proof. Let $A = \mathrm{CYL}_0 \cup \mathrm{CYL}_1$ and $B = \mathrm{CYL}_0$.

Let $\mathcal{L} = \{A \cup C \mid \mathrm{card}(C) < \infty\} \cup \{B \cup C \mid \mathrm{card}(C) < \infty\}$.

$\mathcal{L} \notin \mathbf{LEquQ}^n\mathbf{Ex}$ follows from Proposition 55.

We now show that $\mathcal{L} \in \mathbf{ResSubQ}^1\mathbf{Ex}$. The learner asks once, the query whether A is a subset of the input language. If yes, then the learner outputs, on input σ, a (standard) grammar for $A \cup \mathrm{content}(\sigma)$. If no, then the learner outputs, on input σ, a (standard) grammar for $B \cup \mathrm{content}(\sigma)$. It is easy to verify that the above learner $\mathbf{ResSubQ}^1\mathbf{Ex}$-identifies $\mathcal{L}$.

One can similarly show that $\mathcal{L} \in \mathbf{ResNC}^1\mathbf{Ex}$. The learner, on positive data σ, outputs a grammar for $A \cup \mathrm{content}(\sigma)$ or $B \cup \mathrm{content}(\sigma)$, based on whether there was ever a no answer/counterexample given to the learner earlier. □

Above proof also shows $\mathbf{BNC}^1\mathbf{Ex} - \mathbf{LEquQ}^n\mathbf{Bc}^* \neq \emptyset$.

We now turn our attention to diagonalization against superset queries.

First we show that, if unbounded finite number of errors in almost all conjectures is allowed for **Bc**-learners, then no finite number of superset queries (even unbounded) receiving least counterexamples helps to learn more than what just regular $\mathbf{Bc}^*$-learners can do. In particular, this result will limit our search of separations of types of learning using bounded number of superset queries from other types of learning only to the cases when the latter types do not allow unbounded number of errors in the correct conjectures.

Theorem 57. $\mathbf{LSupQ}^*\mathbf{Bc}^* \subseteq \mathbf{TxtBc}^*$.

Proof. Suppose $\mathbf{M}$ $\mathbf{LSupQ}^*\mathbf{Bc}^*$-identifies a class $\mathcal{L}$. Let $\mathbf{M}'$ be defined as follows. On input $T[m]$, output a grammar for the language defined as follows:

$$L_m = \bigcup_{s \in N} S_m^s$$

In above, $S_m^s = W_{\mathbf{M}(T[m]),s}$, where answers to questions W_j by $\mathbf{M}$ are given as follows:

If $T[m] \subseteq W_{j,s}$, then answer yes.

If $T[m] \not\subseteq W_{j,s}$, then answer no, with $\min(T[m] - W_{j,s})$ as the counterexample.

Let m' be large enough so that if the answers to questions of $\mathbf{M}$ on prefixes of $T[m']$ are correct (for input language being $\mathrm{content}(T)$), then all the questions have been asked by the time $\mathbf{M}$ sees $T[m']$, and for all queried languages W_j, if $\mathrm{content}(T) - W_j \neq \emptyset$, then $\min(\mathrm{content}(T[m']) - W_j) = \min(\mathrm{content}(T) - W_j)$. It is then easy to see that for all $m \geqslant m'$, for all but finitely many s, the simulation of $\mathbf{M}$ as in computation of S_m^s would be correct. Thus, for all $m \geqslant m'$, $\mathbf{M}'(T[m])$ conjectures a language which is a finite variant of $W_{\mathbf{M}(T[m])}$. This is so since L_m would contain $W_{\mathbf{M}(T[m])}$ (with counterexamples to $\mathbf{M}$ being the least ones, if any) and S_m^s, for finitely many s, where some of the answers given to $\mathbf{M}$ may be wrong due to $T[m]$ being a subset of W_j but not $W_{j,s}$, for some query W_j.

Theorem follows. □

The above result is used to derive the following corollary, demonstrating that **Ex**-learners making just one subset or equivalence query, or getting just one bounded negative counterexample to conjectures can sometimes do better than any $\mathbf{Bc}^*$-learner using any finite (unbounded) number of superset queries and receiving least counterexamples.

Corollary 58. $\mathbf{SubQ}^1\mathbf{Ex} \cap \mathbf{EquQ}^1\mathbf{Ex} \cap \mathbf{NC}^1\mathbf{Ex} \cap \mathbf{BNC}^1\mathbf{Ex} - \mathbf{LSupQ}^*\mathbf{Bc}^* \neq \emptyset$.

Proof. Let $\mathcal{L} = \{L \mid L = N \text{ or } (\exists S \mid \text{card}(S) < \infty)[L = \{2x \mid x \in N\} \cup S]\}$. It is easy to verify that $\mathcal{L} \in \mathbf{SubQ}^1\mathbf{Ex} \cap \mathbf{EquQ}^1\mathbf{Ex} \cap \mathbf{BNC}^1\mathbf{Ex} \cap \mathbf{NC}^1\mathbf{Ex}$. However $\mathcal{L} \notin \mathbf{TxtBc}^*$ (by Proposition 11), and hence $\mathcal{L} \notin \mathbf{LSupQ}^*\mathbf{Bc}^*$ by Theorem 57. □

9. Other types of queries versus subset queries and NC

We have already shown that $\mathbf{ResEquQ}^1\mathbf{Ex} - \mathbf{NC}^n\mathbf{Bc}^* \neq \emptyset$ (see Theorem 43).

The following theorem will be useful for the diagonalization $\mathbf{ResEquQ}^1\mathbf{Ex} - \mathbf{LSubQ}^*\mathbf{Bc}^m \neq \emptyset$, as well as for $\mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{LSubQ}^*\mathbf{Bc}^m \neq \emptyset$ and $\mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{LNC}^n\mathbf{Bc}^m \neq \emptyset$ (Corollary 61 below). Note that diagonalization, with superset queries on the positive side, cannot be improved due to $\mathbf{LSupQ}^*\mathbf{Bc}^* \subseteq \mathbf{TxtBc}^*$ (see Theorem 57). However diagonalization, with equivalence queries on the positive side, can be somewhat improved, based on type of counterexamples received for the queries.

Theorem 59. $\mathbf{ResEquQ}^1\mathbf{Ex} \cap \mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{NCBc} \neq \emptyset$.

Proof. Let

$\mathcal{L}_1 = \{L \mid (\exists e)[L = \{\langle 0, e\rangle\} \cup \{\langle 1, x\rangle \mid x \in W_e\}]\}$.
$\mathcal{L}_2 = \{L \mid (\exists e)[$
$\quad \{\langle 0, e\rangle\} \subseteq L \subseteq \{\langle 0, e\rangle\} \cup \text{CYL}_1$ and W_e is finite, and
$\quad\quad (\exists w)[L \cap \{\langle 1, 2w\rangle, \langle 1, 2w+1\rangle\} = \emptyset$ and
$\quad\quad\quad (\forall x < w)[L \cap \{\langle 1, 2x\rangle, \langle 1, 2x+1\rangle\} \neq \emptyset]$, and
$\quad\quad\quad (\forall y > w)[\langle 1, 2y\rangle \in L \ \wedge \ \langle 1, 2y+1\rangle \notin L]]$.
$\quad]\}$
Let $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$.

It is easy to verify that $\mathcal{L}_2 \in \mathbf{TxtEx}$ (on input language L, one just needs to search for the least w such that both $\langle 1, 2w\rangle, \langle 1, 2w+1\rangle$ do not belong to the input language; this information along with the unique e such that $\langle 0, e\rangle \in L$ and $\{\langle 1, x\rangle \mid x < w, \langle 1, x\rangle \in L\}$, is enough to determine $L \in \mathcal{L}$).

It can be shown that $\mathcal{L} \notin \mathbf{NCBc}$, by using essentially the same diagonalization proof as used for showing $\mathbf{InfEx} - \mathbf{NCBc} \neq \emptyset$ in [19]. (Here **InfEx** is a notion of learning from informants [15], where both positive and negative data is given to the learner).

To show that $\mathcal{L} \in \mathbf{ResEquQ}^1\mathbf{Ex}$ (or $\mathbf{ResSupQ}^1\mathbf{Ex}$) one can first obtain e from the input text, and then ask the query whether $\{\langle 0, e\rangle\} \cup \{\langle 1, x\rangle \mid x \in W_e\}$ is equivalent to (superset of) the input language. If yes, then we know the input language. If not, then the input language belongs to $\mathcal{L}_2$, and thus one can use **TxtEx**-identification strategy to identify it. □

Corollary 60. *Suppose $n \in N$. Then,*

(a) $(\mathbf{ResEquQ}^1\mathbf{Ex} \cap \mathbf{ResSupQ}^1\mathbf{Ex}) - \mathbf{LSubQ}^*\mathbf{Bc} \neq \emptyset$.
(b) $(\mathbf{ResEquQ}^1\mathbf{Ex} \cap \mathbf{ResSupQ}^1\mathbf{Ex}) - \mathbf{LSubQ}^*\mathbf{Ex}^* \neq \emptyset$.

Proof. Proposition follows from Theorem 59, Proposition 40 and the fact that $\mathbf{LNCEx}^* \subseteq \mathbf{LNCBc} = \mathbf{NCBc}$ (see [19]). □

The following corollary demonstrates that just one restricted equivalence or superset query made by an **Ex**-learner can sometimes do better than any finite (unbounded) number of subset queries receiving least counterexamples or any bounded number of least negative counterexamples to conjectures used by a $\mathbf{Bc}^m$-learner.

Corollary 61. *For all* $n, m \in N$

(a) $\mathbf{ResEquQ}^1\mathbf{Ex} - \mathbf{LSubQ}^*\mathbf{Bc}^m \neq \emptyset$.
(b) $\mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{LSubQ}^*\mathbf{Bc}^m \neq \emptyset$.
(c) $\mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{LNC}^n\mathbf{Bc}^m \neq \emptyset$.

Proof. (a) We show that $\mathbf{ResEquQ}^1\mathbf{Ex} - \mathbf{SubQ}^*\mathbf{Bc}^m \neq \emptyset$. Part (a) would then follow using Proposition 41.

Note that (i) $Q \subseteq L$ iff $cyl(Q) \subseteq cyl(L)$;
(ii) $Q' \subseteq cyl(L)$ iff $\{x \mid (\exists y)[\langle x, y\rangle \in Q']\} \subseteq L$.

Thus, queries/answers for $\mathbf{SubQ}^*\mathbf{Bc}^m$ learner for $\mathcal{L}$ can be converted to the queries/answers for $\mathbf{SubQ}^*\mathbf{Bc}^m$ learner for $cyl(\mathcal{L})$ and vice-versa.

Furthermore,
(iii) $X = L$ iff $cyl(X) = cyl(L)$;
(iv) $X =^m cyl(L)$ iff $\{x \mid \text{card}(\{y \mid \langle x, y\rangle \in X\} > 2m + 1)\} = L$.

Thus, grammars for L can be converted to grammars for $cyl(L)$ and grammars for m-variant of $cyl(L)$ can be converted to grammars for L.

Using above, it is easy to see that $\mathcal{L} \in \mathbf{SubQ}^n\mathbf{Bc}$ iff $cyl(\mathcal{L}) \in \mathbf{SubQ}^n\mathbf{Bc}$ iff $cyl(\mathcal{L}) \in \mathbf{SubQ}^n\mathbf{Bc}^m$.

Similarly, it can be shown that $\mathcal{L} \in \mathbf{ResEquQ}^1\mathbf{Ex}$ iff $cyl(\mathcal{L}) \in \mathbf{ResEquQ}^1\mathbf{Ex}$.

Thus, it follows that from Corollary 60, that $\mathbf{ResEquQ}^1\mathbf{Ex} - \mathbf{SubQ}^*\mathbf{Bc}^m \neq \emptyset$.

Part (a) now follows using Proposition 41.

(b) Can be proven in a way similar to part (a).

(c) Can be proven by using a slight modification of expanded proof of part (b) (i.e., including the proof for the portion from [19]), where instead of diagonalization against **LSubQ** -query, one diagonalizes against the conjectures, forcing n of them to have counterexamples. We omit the details. □

In contrast to a number of separations established above, as well as Theorem 63 below, our next theorem shows that n restricted equivalence queries made by $\mathbf{Bc}^*$-learners can be simulated by n subset queries. Here, lack of the power of equivalence queries is compensated by possibility of unbounded number of errors in the correct conjectures.

Theorem 62. *For all* $n \in N$, $\mathbf{ResEquQ}^n\mathbf{Bc}^* \subseteq \mathbf{ResSubQ}^n\mathbf{Bc}^*$.

Proof. Suppose $\mathbf{M}$ $\mathbf{ResEquQ}^n\mathbf{Bc}^*$-identifies a class $\mathcal{L}$. Let $\mathbf{M}'$ be defined as follows.

If $\text{content}(T[m]) = \emptyset$, then $\mathbf{M}'(T[m])$ outputs a standard grammar for $\emptyset$. Otherwise, on input $T[m]$, $\mathbf{M}'$ simulates $\mathbf{M}$, asking the same queries as $\mathbf{M}$ does on prefixes of $T[m]$. In the simulation, the answers given to the queries by $\mathbf{M}$ is always no. Suppose the queried languages are (in order of query being made) $W_{j_0}, W_{j_1}, \ldots, W_{j_k}$, where $k < n$. Let p_m denote the final conjecture by $\mathbf{M}$ based on above simulation.

Let,

$$x^i_m = \begin{cases} -1, & \text{if answer to subset query for } W_{j_i} \text{ was no;} \\ \min(\text{content}(T[m]) - W_{j_i,m}), & \text{if answer to subset query for } W_{j_i} \text{ was yes and content}(T[m]) - W_{j_i,m} \neq \emptyset; \\ \min(\text{content}(T[m])), & \text{otherwise.} \end{cases}$$

For the following, we take $-1 \notin W_{j_i}$ (this is for ease of presentation). Then, $\mathbf{M}'$ on $T[m]$, outputs a program for the following language:

$$L_m = \bigcup_{s \in N, (\forall i \leqslant k)[x^i_m \notin W_{j_i,s}]} [W_{p_m,s}] \cup \bigcup_{s \in N, r = \min(\{i \mid x^i_m \in W_{j_i,s}\})} [W_{j_r,s}]$$

Now suppose T is a text for $L \in \mathcal{L}$. Consider the following cases.

Case 1: For all $r \leqslant k$, $W_{j_r} \neq L$.

Let

$$y^i_m = \begin{cases} -1 & \text{if answer to subset query for } W_{j_i} \text{ was no;} \\ \min(\text{content}(T) - W_{j_i}) & \text{if answer to subset query for } W_{j_i} \text{ was yes.} \end{cases}$$

Note that, for all but finitely many m, $x^i_m = y^i_m$. Thus, for all but finitely many m, the language L_m defined above is W_{p_m}. Hence, $\mathbf{M}'$ $\mathbf{ResSubQ}^n\mathbf{Bc}^*$-identifies L on text T.

Case 2: $W_{j_r} = L$, for some $r \leqslant k$.

Then choose the minimal such r. For $i < r$, define

$$y^i_m = \begin{cases} -1 & \text{if answer to subset query for } W_{j_i} \text{ was no.} \\ \min(\text{content}(T) - W_{j_i}) & \text{if answer to subset query for } W_{j_i} \text{ was yes.} \end{cases}$$

Now, for $i < r$, for all but finitely many m, $y^i_m = x^i_m$. Moreover, $x^r_m \neq -1$, and $x^r_m \in \text{content}(T) = W_{j_r}$ for all m.

Thus, for all but finitely many m, for all but finitely many s, $[\neg(\forall i \leqslant k)[x^i_m \notin W_{j_i,s}]]$. Moreover, for all but finitely many m, for all but finitely many s, $\min(\{i \mid x^i_m \in W_{j_i,s}\})$ would be r.

Thus, for all but finitely many m, $L_m =^* W_{j_r}$ (as L_m would contain W_{j_r} and some finite sets due to "finitely many s" for which $(\forall i \leqslant k)[x^i_m \notin W_{j_i,s}]$, holds, or $\min(\{i \mid x^i_m \in W_{j_i,s}\}) \neq r$ holds).

Hence, $\mathbf{M}'$ $\mathbf{ResSubQ}^n\mathbf{Bc}^*$-identifies L on text T.

Theorem follows from above analysis. □

Now we show that **Ex**-learners making just two equivalence queries can sometimes do better than any **Bc***-learner making unbounded finite number of subset queries receiving least counterexamples.

Theorem 63. $\mathbf{EquQ}^2\mathbf{Ex} - \mathbf{LSubQ}^*\mathbf{Bc}^* \neq \emptyset$.

Proof. Consider the following class:

$A = \bigcup_{j \in K} \mathrm{CYL}_j$.
$B_i = A \cup \mathrm{CYL}_i$.
$B_i^k = A \cup \{\langle i, x\rangle \mid x \leqslant k\}$.
$\mathcal{L} = \{A\} \cup \{B_i \mid i \notin K\} \cup \{B_i^k \mid i \notin K, k \in N\}$.

Intuitively, an equivalence query with A allows one to know if the input set is A or B_i/B_i^k, along with knowing i (using the counterexample). This, allows $\mathbf{EquQ}^2\mathbf{Ex}$- learnability of $\mathcal{L}$. On the other hand, it can be shown that subset queries are not able to get the crucial information about which i is used above. See details below.

Claim 64. $\mathcal{L} \in \mathbf{EquQ}^2\mathbf{Ex}$.

Proof. A learner first asks equivalence query for language A. If the answer is yes, then we are done. Otherwise suppose $\langle i, j\rangle$ is the counterexample. Then the learner asks equivalence query for the language B_i. If the answer is yes, then we are done. Otherwise, the language must be B_i^k, for some k. This k can be easily determined in the limit, by checking for $\max(\{x \mid \langle i, x\rangle \in \mathrm{content}(T)\})$, where T is the input text. Claim follows.

Claim 65. $\mathcal{L} \notin \mathbf{LSubQ}^*\mathbf{Bc}^*$.

Proof. Suppose by way of contradiction $\mathbf{M}$ $\mathbf{LSubQ}^*\mathbf{Bc}^*$-identifies $\mathcal{L}$. Let σ be a $\mathbf{LSubQ}^*\mathbf{Bc}^*$-locking sequence for $\mathbf{M}$ on A. That is, $\mathrm{content}(\sigma) \subseteq A$, and $\mathbf{M}$ does not ask any more questions on any extension σ' of σ, with $\mathrm{content}(\sigma') \subseteq A$, as long as questions Q of $\mathbf{M}$ on prefixes of σ are answered as follows:

(i) If $Q \subseteq A$, then answer yes;

(ii) If $Q \not\subseteq A$, then answer no, and give $\min(Q - A)$ as a counterexample;

($\mathbf{M}$ also needs to output grammars for finite variant of A on extensions of σ, but that is not important for following.)

Let $S = \{i \mid \langle i, x\rangle$ is given as a counterexample to $\mathbf{M}$ in the above process on some query on a prefix of $\sigma\}$.

Now we claim that there exists a $i \notin K \cup S$, such that for any $\sigma' \supseteq \sigma$, $\mathrm{content}(\sigma') \subseteq B_i$, $\mathbf{M}(\sigma')$ does not ask a question. (If not, then clearly one can show $\overline{K} - S$ to be r.e., by enumerating all $i \notin S$, such that $\mathbf{M}$ asks a question on some $\sigma' \supseteq \sigma$, with $\mathrm{content}(\sigma') \subseteq B_i$. A contradiction to K being non-recursive.)

Thus, let i be such that $\mathbf{M}$ does not ask a question on any $\sigma' \supseteq \sigma$ such that $\mathrm{content}(\sigma') \subseteq B_i$. Thus, $\mathbf{M}$ now needs to $\mathbf{Bc}^*$-identify B_i as well as B_i^k, without asking any more questions. This is impossible by Proposition 11.

Theorem follows from the above claims. □

10. Learning via superset queries versus learning via equivalence queries

Note that by Corollary 58, $\mathbf{ResEquQ}^1\mathbf{Ex} - \mathbf{LSupQ}^*\mathbf{Bc}^* \neq \emptyset$. We now consider the diagonalization from superset queries against equivalence queries. Note that $\mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{LEquQ}^n\mathbf{Bc}^t \neq \emptyset$ cannot be improved to having $\mathbf{Bc}^*$ on the RHS (as $\mathbf{LSupQ}^*\mathbf{Bc}^* \subseteq \mathbf{TxtBc}^*$, Theorem 57) or to having $*$-number of equivalence queries (as $\mathcal{E} \in \mathbf{ResEquQ}^*\mathbf{Ex}$, Proposition 54).

Theorem 66. *For all $n \in N$, there exists a $\mathcal{L}$ such that*
(a) for all $t \in N$, $\mathcal{L} \notin \mathbf{LEquQ}^n\mathbf{Bc}^t$;
(b) $\mathcal{L} \notin \mathbf{LEquQ}^n\mathbf{Ex}^*$;
(c) $\mathcal{L} \in \mathbf{ResSupQ}^1\mathbf{Ex}$.

Proof. Consider the following class of languages.

$\mathcal{L} = \{L \mid (\exists e > 0)[$
1. $\langle e, 0\rangle > \langle 0, n\rangle$,
2. $(L - \mathrm{CYL}_e) \subseteq \{\langle 0, x\rangle \mid x < n\}$.
3. Either
   3.1 W_e is infinite and $L \cap \mathrm{CYL}_e = \{\langle e, x\rangle \mid x \in W_e\}$.
   or
   3.2 W_e is finite, and $(\exists w)[$
   $L \cap \{\langle e, 2w\rangle, \langle e, 2w+1\rangle\} = \emptyset$ and
   $(\forall x < w)[L \cap \{\langle e, 2x\rangle, \langle e, 2x+1\rangle\} \neq \emptyset]$, and
   $(\forall y > w)[\langle e, 2y\rangle \in L \ \wedge \ \langle e, 2y+1\rangle \notin L]]$.
   ]}

It is easy to verify that $\mathcal{L} \in \mathbf{SupQ}^1\mathbf{Ex}$. One first waits for an $e > 0$, such that the input contains an element from CYL_e. Then one queries whether $\mathrm{CYL}_0 \cup \{\langle e, x\rangle \mid x \in W_e\}$ is a superset of the input language. If the answer is yes, then the clause 3.1, in the definition of $\mathcal{L}$ must have applied. If the answer is no, then the clause 3.2 in the definition of $\mathcal{L}$ must have applied. In both cases, it is easy to determine the input language using the text.

We now consider the diagonalization against $\mathbf{LEquQ}^n\mathbf{Bc}^t$ and $\mathbf{LEquQ}^n\mathbf{Ex}^*$. Intuitively, elements $< \langle 0, n\rangle$, would be used to answer equivalence queries by a supposed $\mathbf{LEquQ}^n\mathbf{Bc}^t$-learner ($\mathbf{LEquQ}^n\mathbf{Ex}^*$-learner) for $\mathcal{L}$. After the final query is made, this would be used along with Lemma 20 (Corollary 21) to get a diagonalization. We now proceed formally.

Claim 67. *(a) For all $t \in N$, $\mathcal{L} \notin \mathbf{LEquQ}^n\mathbf{Bc}^t$.*
(b) $\mathcal{L} \notin \mathbf{LEquQ}^n\mathbf{Ex}^*$.

Proof. We only show part (a). Part (b) can be shown using Corollary 21 instead of using Lemma 20.

Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{LEquQ}^n\mathbf{Bc}^t$-identifies $\mathcal{L}$. For each $e \in N$, such that $\langle e, 0\rangle > \langle 0, n\rangle$, we will define below σ_i^e and C_i^e. It will be the case that $C_i^e \subseteq \{\langle 0, x\rangle \mid x < i\}$. Initially, let $C_0^e = \emptyset$ and $\sigma_0^e = \Lambda$.

Inductively define $\sigma_{i+1}^e, C_{i+1}^e$, for $i < n$ as follows.
(* The construction is non-effective. However, one can determine some things limit-effectively in e, see below. *)
(* Following invariants will be satisfied:
   (a) $C_i^e \subseteq \{\langle 0, x\rangle \mid x < i\}$.
   (b) $C_i^e \subseteq \mathrm{content}(\sigma_i^e) \subseteq C_i^e \cup \mathrm{CYL}_e$.
   (c) $\mathbf{M}$ has asked at least i queries on σ_i^e.
   (d) For $\langle e, 0\rangle > \langle 0, n\rangle$, answers given to queries by $\mathbf{M}$ are consistent with any input language L such that $\mathrm{content}(\sigma_i^e) \subseteq L \subseteq C_i^e \cup \{\langle 0, x\rangle \mid i \leqslant x < n\} \cup \mathrm{CYL}_e$.
   *)

1. Check if there exists a $\sigma \supseteq \sigma_i^e$ such that $\text{content}(\sigma) \subseteq C_i^e \cup \text{CYL}_e$ and $\mathbf{M}$ asks a query on σ.
   If there is no such σ, then σ_j^e, C_j^e for $j > i$ do not get defined.
   If there exists such a σ, then fix one such σ and proceed as follows.
2. Suppose Q was the language queried.
   Let $C_{i+1}^e = C_i^e$, if $\langle 0, i\rangle \in Q$.
   Let $C_{i+1}^e = C_i^e \cup \{\langle 0, i\rangle\}$, if $\langle 0, i\rangle \notin Q$.
   Answer the query as no, with the least counterexample being the least element in $Q\mathbf{\Delta}C_{i+1}^e$ (note that such an element $\leqslant \langle 0, i\rangle$ exists by definition of C_{i+1}^e above, as $\langle 0, i\rangle \in Q\mathbf{\Delta}C_{i+1}^e$).
   Let σ_{i+1}^e be an extension of σ such that $\text{content}(\sigma_{i+1}^e) = \text{content}(\sigma) \cup C_{i+1}^e$.
   (* We assume without loss of generality that if $\sigma \subset \sigma' \subseteq \sigma_{i+1}$, then $\mathbf{M}$ does not ask any questions. If not, then one can just delay these questions beyond σ_{i+1}, without effecting this construction. *)

End

It is easy to verify that the invariants are satisfied. Let m be largest such that σ_m^e is defined. Further note that $\mathbf{M}$ does not ask any more queries on any T which extends σ_m^e and $\text{content}(T) \subseteq \text{content}(\sigma_m^e) \cup \text{CYL}_e$.

It is easy to verify that one can obtain σ_m^e from e limit-recursively. That is, there exists a recursive function g mapping $N \times N$ to SEQ such that $\lim_{s\to\infty} g(e,s)$ converges to σ_m^e.

Furthermore, there exists a recursive function h such that $\lim_{s\to\infty} h(e,s)\downarrow$ and $\mathbf{M}_{\lim_{s\to\infty} h(e,s)}$ $\mathbf{TxtBc}^t$-identifies all L such that $\mathbf{M}$ $\mathbf{LEquQ}^n\mathbf{Bc}^t$ identifies L and $\text{content}(\sigma_m^e) \subseteq L \subseteq \text{content}(\sigma_m^e) \cup \text{CYL}_e$ (Here $\mathbf{M}_0, \mathbf{M}_1, \ldots$, denote a listing of all $\mathbf{TxtBc}^t$-learning machines).

Now, let F be as in Lemma 20.

Intuitively, we would like to use Lemma 20, for the finite S (as in the lemma) being $\text{content}(\sigma_m^e)$ and the machine (as in the lemma) being $\mathbf{M}_{\lim_{s\to\infty} h(e,s)}$. However, as these values can only be obtained in the limit, we need to appropriately modify the finite set S, to handle the elements that W_e (defined below) may have enumerated before knowing the final value of σ_m^e and $\mathbf{M}_{\lim_{s\to\infty} h(e,s)}$.

For $s \in N$, let $a_s^e = \max(\{s' < s \mid g(e,s') \neq g(e,s'+1) \text{ or } h(e,s') \neq h(e,s'+1)\})$. Intuitively, a_s^e denotes the last time $s' < s$, such that a change in $g(e,\cdot)$ or $h(e,\cdot)$ was observed.

By Kleene's recursion theorem [28] there exists an e such that $\langle e, 0\rangle > \langle 0, n\rangle$ and $W_e = \bigcup_{s\in N} X_s$, where

$X_s = W_{F(e,A_s,\mathbf{M}_{h(e,s)}),s}$, and $A_s = \text{content}(g(e,s)) \cup \bigcup_{s' \leqslant a_s^e} X_{s'}$.

Intuitively, we want W_e to simulate $W_{F(e,\text{content}(\sigma_m^e)\cup S,\mathbf{M}_{\lim s\to\infty h(e,s)})}$, where S is the finite stuff which W_e had enumerated due to earlier inaccurate value of σ_m^e and $\mathbf{M}_{\lim s\to\infty h(e,s)}$, it may have tried before using the correct values. Here, note that $\lim_{s\to\infty} A_s$ would contain $\text{content}(\sigma_m^e)$ and whatever X_s's W_e may have enumerated before knowing the final value of $\text{content}(\sigma_m^e)$ and $\mathbf{M}_{\lim_{s\to\infty} h(e,s)}$.

It now follows from Lemma 20 that $\mathbf{M}_{\lim_{s\to\infty} h(e,s)}$ does not $\mathbf{TxtBc}^t$-identify some language $L \in \mathcal{L}$ such that $\text{content}(\sigma_m^e) \subseteq L$. Thus, $\mathbf{M}$ does not $\mathbf{LEquQ}^n\mathbf{Bc}^t$-identify L and hence $\mathcal{L}$. □

11. Anomaly hierarchy

In this section, we give the anomaly hierarchy for the various query learning criteria.

Proposition 68. *(Based on [10]) Suppose X is an infinite language, $S \subseteq X$ and $X - S$ is infinite. Let, $\mathcal{L}_n = \{L \mid S \subseteq L \subseteq X \text{ and } \mathrm{card}(X - L) \leqslant n\}$. Then,*

(a) $\mathcal{L}_{n+1} \notin \mathbf{TxtEx}^n$.

(b) $\mathcal{L}_{2n+1} \notin \mathbf{TxtBc}^n$.

We first consider superset queries.

Theorem 69. *Let $\mathcal{L}_n = \{L \mid L =^n N\}$. Then,*

(a) $\mathcal{L}_n \in \mathbf{TxtEx}^n$.

(b) $\mathcal{L}_{n+1} \notin \mathbf{LSupQ}^*\mathbf{Ex}^n$.

(c) $\mathcal{L}_{2n+1} \notin \mathbf{LSupQ}^*\mathbf{Bc}^n$.

Proof. (a) is straightforward, as a learner just needs to output a grammar for N.

(b) Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{LSupQ}^*\mathbf{Ex}^n$-identifies $\mathcal{L}_{n+1}$. Consider the learnability of N by $\mathbf{M}$. A query about language Q is answered as follows: if $Q = N$, then answer yes. Otherwise answer no and return the least element in $N - Q$. Let σ be such that $\mathbf{M}$ does not ask any new queries on any extension of σ, as long as answers are given as above on queries made on initial segments of σ. (Note that such a σ exists, since otherwise $\mathbf{M}$ on some text for N makes infinitely many queries). Let S be the collection of all elements which are given as counterexamples to queries answered as no above.

Now, one can easily modify $\mathbf{M}$ to $\mathbf{TxtEx}^n$-identify the class $\mathcal{L} = \{L \mid L =^{n+1} N \text{ and } (S \cup \mathrm{content}(\sigma)) \subseteq L\}$. However this is not possible by Proposition 68(a). This proves part (b).

Part (c) can be similarly proved by using Proposition 68(b). □

We next consider equivalence queries. Note that as $\mathcal{E} \in \mathbf{ResEquQ}^*\mathbf{Ex}$ (Proposition 54), we can only consider the hierarchy for bounded number of queries.

Theorem 70. *Fix $m \in N$. Let $X = \{x \mid x \geqslant m\}$. Let $\mathcal{L}_n = \{L \mid \mathrm{card}(X - L) \leqslant n\}$. Then,*

(a) $\mathcal{L}_n \in \mathbf{TxtEx}^n$.

(b) $\mathcal{L}_{n+1} \notin \mathbf{LEquQ}^m\mathbf{Ex}^n$.

(c) $\mathcal{L}_{2n+1} \notin \mathbf{LEquQ}^m\mathbf{Bc}^n$.

Proof. Part (a) can be easily shown by outputting on input text T, a grammar (in the limit) for $(\mathrm{content}(T) \cap \{x \mid x < m\}) \cup X$.

(b) Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{LEquQ}^m\mathbf{Ex}^n$-identifies $\mathcal{L}_{n+1}$.

Let $\tau_0 = \Lambda$. For $i < m$, τ_{i+1} is defined as follows.

If there does not exist a $\sigma \supseteq \tau_i$ such that $\mathrm{content}(\sigma) \subseteq \mathrm{content}(\tau_i) \cup X$, and $\mathbf{M}$ asks a query on input σ, then let $\tau_{i+1} = \tau_i$ (note that in this case, by iterating the above process, we would also have $\tau_m = \tau_i$).

On the other hand, if there exists $\sigma \supseteq \tau_i$ such that $\mathrm{content}(\sigma) \subseteq \mathrm{content}(\tau_i) \cup X$, and $\mathbf{M}$ asks a query on input σ, then fix smallest such σ. Suppose the query is about language Q_i. Let $\tau_{i+1} = \sigma$ if $i \in Q_i$; otherwise let $\tau_{i+1} = \sigma \diamond i$. Note that $i \in (Q_i \mathbf{\Delta} \mathrm{content}(\tau_{i+1}))$. Answer the query (at σ) as no, and give counterexample as the least element in $Q_i \mathbf{\Delta} (\mathrm{content}(\tau_{i+1}) \cup X)$. Note that there exists such an element $\leqslant i$. We will make sure that only elements $> i$ or elements already in $\mathrm{content}(\sigma)$ would be used for extending τ_{i+1}, thus maintaining the correctness of the answers given.

Now note that **M** does not ask any further queries on any text extending τ_m for the language content$(\tau_m) \cup X$ (either it has already asked m questions or we had explicitly checked above that **M** does not ask any further questions due to non-existence of σ in the definition of τ_m above). Thus, we can easily modify **M** to **TxtEx**^{n}-identify all languages in $\{L \mid \text{content}(\tau_n) \subseteq L \subseteq \text{content}(\tau_n) \cup X$ and $\text{card}(X - L) \leqslant n + 1\}$. However this is not possible by Proposition 68(a). This proves part (b).

Part (c) can be similarly proved by using Proposition 68(b) □

We now consider subset queries. The following theorem shows that we cannot get diagonalizations of form **Ex**$^{2n+1}$ vs **Bc**^{n}, in case of subset queries.

Theorem 71. LSubQ*Ex* $\subseteq$ ResSubQ*Bc .

Proof. We only give a sketch of the proof. As **ResSubQ*Bc = SubQ*Bc = LSubQ*Bc** and **ResSubQ*Ex* = SubQ*Ex* = LSubQ*Ex***, it suffices to show **SubQ*Ex* $\subseteq$ SubQ*Bc**.

Suppose **M SubQ*Ex***-identifies a class $\mathcal{L}$. Then, $\mathbf{M}'$ is defined as follows. On any text T for $L \in \mathcal{L}$, $\mathbf{M}'$ would simulate **M**. Queries of **M** can be easily answered by making the same queries. Moreover, errors of commission of the last conjecture of **M** on T can be removed by detecting them using subset queries. As there are only finitely many errors in the last conjecture, this requires only finitely many subset queries. Errors of ommission can be patched by including content$(T[n])$ in the conjecture made at $T[n]$. Thus, eventually $\mathbf{M}'$ can patch all errors of the last conjecture of **M**. □

Theorem 72. *(a)* **TxtEx**$^{n+1}$ **− LSubQ*Ex^n** $\neq \emptyset$.

(b) **TxtBc**$^{n+1}$ **− LSubQ*Bc**^{n} $\neq \emptyset$.

Proof. (a) As **LSubQ*Ex^n = ResSubQ*Ex^n**, it suffices to show **TxtEx**$^{n+1}$ **− ResSubQ*Ex^n** $\neq \emptyset$.

Let $\mathcal{L} = \{L \mid W_{\min(L)} =^{n+1} L\}$. It is easy to verify that $\mathcal{L} \in$ **TxtEx**$^{n+1}$.

Suppose by way of contradiction **M ResSubQ*Ex^n**-identifies $\mathcal{L}$. Then, by implicit use of Kleene Recursion Theorem [28], there exists an e such that W_e may be defined as follows.

Let σ_0 be a finite sequence containing just one element e. Let $S_0 = \{x \mid x < e\}$. Enumerate e in W_e. Let W_e^s denote W_e enumerated before stage s. We will have the invariant that content$(\sigma_s) = W_e^s$, and $S_s \cap W_e^s = \emptyset$. Go to stage 0.

Stage s

1. Dovetail steps 2 and 3, until one of them succeeds. If step 2 succeeds before step 3, if ever, then go to step 4. If step 3 succeeds before step 2, if ever, then go to step 5. Here we assume that if there exists a query j made on a prefix of σ_s which satisfies: $W_{j,s} \not\subseteq W_e^s$ and $W_{j,s} \cap S_s = \emptyset$, then step 3 succeeds first (i.e., some priority is given to step 3).
2. Search for an extension σ of σ_s such that content$(\sigma) \cap S_s = \emptyset$, and card(content$(\sigma) - W_e^s) \leqslant n + 1$, and either **M** makes a query at σ or $\mathbf{M}(\sigma) \neq \mathbf{M}(\sigma_s)$. Here answers to queries j made by **M** on prefixes of σ are answered yes, iff $W_{j,s} \cap S_s = \emptyset$.
3. Search for a query j made on prefixes of σ_s such that for some $t \geqslant s$, $W_{j,t} \not\subseteq W_e^s$ and $W_{j,t} \cap S_s = \emptyset$ (here answers to queries k made by **M** on prefixes of σ are answered yes, iff $W_{k,s} \cap S_s = \emptyset$).
4. If and when such a σ is found, let $\sigma_{s+1} = \sigma\#$.
   Enumerate content(σ_{s+1}) in W_e.
   Let $S_{s+1} = S_s$.
   Go to stage $s + 1$.

5. Let t be as found in step 3 above.
   Let QS be the set of all the possible queries made on initial segments of σ_s, based on all possible ways of answering the queries.
   Let $S_{s+1} = (S_s \cup \bigcup_{j \in QS} W_{j,t}) - W_e^s$.
   Let $\sigma_{s+1} = \sigma_s$.
   Go to stage $s+1$.

End Stage s

We now consider the following cases.

Case 1: There are only finitely many stages.

Let s be the last stage which is entered. Note that as step 3 does not succeed, all answers given to $\mathbf{M}$ are correct for any input language satisfying content$(\sigma_s) \subseteq L$, $L \cap S_s = \emptyset$, and card$(L -$ content$(\sigma_s)) \leqslant n+1$. Also, all such languages are in $\mathcal{L}$. Furthermore, for any text T (for such L) which extends σ_s, $\mathbf{M}(T)$ does not ask any questions beyond σ_s, and also $\mathbf{M}(T) = \mathbf{M}(\sigma_s)$.

Let $Z = W_{\mathbf{M}(\sigma_s)} -$ content(σ_s). If card$(Z) > n$, then $\mathbf{M}$ does not $\mathbf{ResSubQ}^*\mathbf{Ex}^n$-identify $L = W_e^s$. On the other hand if card$(Z) \leqslant n$, then $\mathbf{M}$ does not $\mathbf{ResSubQ}^*\mathbf{Ex}^n$-identify any $L = W_e^s \cup Y$, with card$(Y) = n+1$ and $Y \cap W_{\mathbf{M}(\sigma_s)} = \emptyset$. It follows that $\mathbf{M}$ does not $\mathbf{ResSubQ}^*\mathbf{Ex}^n$-identify $\mathcal{L}$.

Case 2: There exist infinitely many stages.

We first claim that step 2 must succeed in infinitely many stages. Suppose otherwise. Let s be a stage such that in every stage $t \geqslant s$, step 3 succeeds. Let s' be so large that for any query j asked on initial segments of σ_s, $W_{j,s'} - W_e^s \neq \emptyset$ or $W_j \subseteq W_e^s$. Now, beyond stage s', each time step 5 is executed a new query j would have been chosen. However, as there are only finitely many queries made by $\mathbf{M}$ on prefixes of σ_s, this would imply that there are only finitely many stages.

Thus, step 2 succeeds in infinitely many stages and thus step 4 is executed in infinitely many stages. Let $T = \bigcup_{s \in N} \sigma_s$, and $L =$ content(T). Let r be such that $\mathbf{M}$ does not ask any more queries on T beyond $T[r]$, if all answers on queries on prefixes of $T[r]$ are answered correctly. Let $s > r$ be large enough such that, for each query j made on prefixes of $T[r]$, either $W_j \subseteq L$ or $W_{j,s} \not\subseteq L$. It follows that step 3 cannot succeed beyond stage $s+1$, and all answers given beyond stage $s+1$ are always correct (in stage s some answers given may be wrong, but these are fixed by updating S_{s+1} appropriately). Thus, as step 2 succeeds in almost all stages beyond stage $s+1$, $\mathbf{M}$ makes infinitely many mind changes on text T when the answers are given correctly to the queries. Thus, $\mathbf{M}$ does not $\mathbf{SubQ}^*\mathbf{Ex}^n$-identify $\mathcal{L}$.

Part (b) can be proved similarly by using the class $\mathcal{L} = \{L \mid \text{card}(L) = \infty \text{ and } (\forall^\infty x \in L)[W_x =^{n+1} L]\}$, and modifying the diagonalization of $\mathbf{Bc}^{n+1} - \mathbf{Bc}^n$ in [11]. □

As corollary to theorems shown in this section we have:

Corollary 73. *Suppose* $a \in N \cup \{*\}$, *and* $m, n \in N$.
(a) $\mathbf{SubQ}^a\mathbf{Ex}^n \subset \mathbf{SubQ}^a\mathbf{Ex}^{n+1}$.
$\mathbf{LSubQ}^a\mathbf{Ex}^n \subset \mathbf{LSubQ}^a\mathbf{Ex}^{n+1}$.
$\mathbf{ResSubQ}^a\mathbf{Ex}^n \subset \mathbf{ResSubQ}^a\mathbf{Ex}^{n+1}$.
(b) $\mathbf{SupQ}^a\mathbf{Ex}^n \subset \mathbf{SupQ}^a\mathbf{Ex}^{n+1}$.
$\mathbf{LSupQ}^a\mathbf{Ex}^n \subset \mathbf{LSupQ}^a\mathbf{Ex}^{n+1}$.
$\mathbf{ResSupQ}^a\mathbf{Ex}^n \subset \mathbf{ResSupQ}^a\mathbf{Ex}^{n+1}$.

(c) $\mathbf{EquQ}^m\mathbf{Ex}^n \subset \mathbf{EquQ}^m\mathbf{Ex}^{n+1}$.
$\mathbf{LEquQ}^m\mathbf{Ex}^n \subset \mathbf{LEquQ}^m\mathbf{Ex}^{n+1}$.
$\mathbf{ResEquQ}^m\mathbf{Ex}^n \subset \mathbf{ResEquQ}^m\mathbf{Ex}^{n+1}$.

Similar corollary exists for **Bc**-criteria of learning with **Ex** being replaced by **Bc** in the above.

The proof of $\mathbf{TxtEx}^{2n} \subseteq \mathbf{TxtBc}^n$ (Result from [10]; For proof see Proposition 6.24 of [20]) can also be used to show the following theorem.

Theorem 74. *Suppose* $QS \in \{\mathbf{SubQ}, \mathbf{ResSubQ}, \mathbf{LSubQ}, \mathbf{SupQ}, \mathbf{ResSupQ}, \mathbf{LSupQ}, \mathbf{EquQ}, \mathbf{ResEquQ}, \mathbf{LEquQ}\}$.

Then, $QS\mathbf{Ex}^{2n} \subseteq QS\mathbf{Bc}^n$.

The above theorem (along with earlier proved diagonalizations in this section) resolves the relationship between **Ex** and **Bc** error hierarchies.

12. Conclusion

In this paper, we explored learning classes of recursively enumerable languages from full positive data and bounded number of subset, superset and equivalence queries. We compared capabilities of learning models using different types of queries and counterexamples and obtained hierarchies based on the number and types of counterexamples. Learning languages from full positive data with potentially unbounded number of negative counterexamples to conjectures was explored in [19], where it was shown that all recursively enumerable languages can be learned by $\mathbf{Bc}^1$-learners, but not by any $\mathbf{Ex}^a$-learners or **Bc**-learners.

In case one is allowed to ask unbounded finitely many proper superset queries, then one can learn the class $\mathcal{E}$ as follows. The learner first asks the query 'is N a proper superset of the input language'. If not, then the input language is N. Otherwise, one determines the least $x \notin L$, and searches for an e such that $[W_e \cup \{x\} \supset L]$ is true, but $[W_e \supset L]$ is false (note that such an e and x can be obtained in the limit using the input text). Then, the input language must be W_e. Similarly, one can show that if a learner is allowed to ask unbounded finitely many proper subset queries, then one can learn the class $\mathcal{E}$. We have not discussed yet another popular and natural type of queries considered in literature - membership queries, as a bounded number of such queries trivially does not help in the presence of full positive data. On the other hand, learning languages from full positive data and infinitely many membership queries is equivalent to learning from full positive and negative data (so-called informants) thoroughly explored in literature ([20]). One can also show that infinite number of (superset, subset or equivalence) queries makes it possible to learn any recursively enumerable language (positive data becomes unnecessary in these cases).

The reader may note the following connection to team learning [30]. A query-learner which is allowed to ask n queries can be simulated by a team of 2^n learners: the learners in the team operate based on the 2^n possible answers to the (first) n queries of the query-learner (counterexamples, if needed, can be obtained in the limit using the input text—assuming that answers to queries are correct).

In our research, we concentrated on learning classes of recursively enumerable languages. One might also consider learning from positive data and bounded number of queries for indexed clas-

ses of recursive languages (they include such important classes as regular languages and pattern languages [2]). Some of our results are applicable to indexed classes of recursive languages. Still, further research in this direction might be promising.

13. Acknowledgments

We thank the anonymous referees for several helpful comments.

References

[1] D. Angluin, L. Hellerstein, M. Karpinski, Learning read-once formulas with queries, Journal of the ACM 40 (1) (1993) 185–210.
[2] D. Angluin, Finding patterns common to a set of strings, Journal of Computer and System Sciences 21 (1980) 46–62.
[3] D. Angluin, Learning regular sets from queries and counter-examples, Information and Computation 75 (1987) 87–106.
[4] D. Angluin, Queries and concept learning, Machine Learning 2 (1988) 319–342.
[5] D. Angluin, Queries revisited, in: Algorithmic Learning Theory: Twelfth International Conference (ALT' 2001), Lecture Notes in Artificial Intelligence, vol. 2225, Springer, Berlin, 2001, pp. 12–31.
[6] J. Bā rzdiņš, Two theorems on the limiting synthesis of functions, in: Theory of Algorithms and Programs, vol. 1, Latvian State University, 1974, pp. 82–88, In Russian.
[7] L. Blum, M. Blum, Toward a mathematical theory of inductive inference, Information and Control 28 (1975) 125–155.
[8] G. Baliga, J. Case, S. Jain, Language learning with some negative information, Journal of Computer and System Sciences 51 (5) (1995) 273–285.
[9] J. Case, Periodicity in generations of automata, Mathematical Systems Theory 8 (1974) 15–32.
[10] J. Case, C. Lynes, Machine inductive inference and language identification, in: M. Nielsen, E.M. Schmidt (Eds.), Proceedings of the 9th International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 140, Springer, Berlin, 1982, pp. 107–115.
[11] J. Case, C. Smith, Comparison of identification criteria for machine inductive inference, Theoretical Computer Science 25 (1983) 193–220.
[12] L. Fortnow, W. Gasarch, S. Jain, E. Kinber, M. Kummer, S. Kurtz, M. Pleszkoch, T. Slaman, R. Solovay, F. Stephan, Extremes in the degrees of inferability, Annals of Pure and Applied Logic 66 (1994) 231–276.
[13] M. Fulk, Prudence and other conditions on formal language learning, Information and Computation 85 (1990) 1–11.
[14] W. Gasarch, G. Martin, Bounded Queries in Recursion Theory, Birkhauser, 1998.
[15] E.M. Gold, Language identification in the limit, Information and Control 10 (1967) 447–474.
[16] W. Gasarch, M. Pleszkoch. Learning via queries to an oracle, in: R. Rivest, D. Haussler, M. Warmuth (Eds.), Proceedings of the Second Annual Workshop on Computational Learning Theory, Morgan Kaufmann, 1989, pp. 214–229.
[17] W. Gasarch, C. Smith, Learning via queries, Journal of the ACM (1991) 649–674.
[18] O. Ibarra, T. Jiang, Learning regular languages from counterexamples, in: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann, 1988, pp. 337–351.
[19] S. Jain, E. Kinber, Learning languages from positive data and negative counterexamples, in: Shai Ben-David, John Case, Akira Maruoka (Eds.), Algorithmic Learning Theory: Fifteenth International Conference (ALT' 2004), Lecture Notes in Artificial Intelligence, vol. 3244, Springer, Berlin, 2004, pp. 54–68.
[20] S. Jain, D. Osherson, J. Royer, A. Sharma, Systems that Learn: An Introduction to Learning Theory, second ed., MIT Press, Cambridge, MA, 1999.
[21] E. Kinber, Learning a class of regular expressions via restricted subset queries, in: K. Jantke (Ed.), Analogical and Inductive Inference, Proceedings of the Third International Workshop, Lecture Notes in Artificial Intelligence, vol. 642, Springer, Berlin, 1992, pp. 232–243.

[22] S. Lange, J. Nessel, S. Zilles, Learning languages with queries, in: Proceedings of Treffen der GI-Fachgruppe Maschinelles Lernen (FGML), Learning Lab Lower Saxony, Hannover, Germany, 2002, pp. 92–99.
[23] S. Lange, S. Zilles, Comparison of query learning and Gold-style learning in dependence of the hypothesis space, in: Shai Ben-David, John Case, Akira Maruoka (Eds.), Algorithmic Learning Theory: Fifteenth International Conference (ALT' 2004), Lecture Notes in Artificial Intelligence, vol. 3244, Springer, Berlin, 2004, pp. 99–113.
[24] S. Lange, S. Zilles, Replacing limit learners with equally powerful one-shot query learners, in: John Shawe-Taylor, Yoram Singer (Eds.), Proceedings of the Seventeenth Annual Conference on Computational Learning Theory, Lecture Notes in Artificial Intelligence, vol. 3120, Springer, Berlin, 2004, pp. 155–169.
[25] J. Nessel, S. Lange, Learning erasing pattern languages with queries, in: Algorithmic Learning Theory: Eleventh International Conference (ALT' 2000), of Lecture Notes in Artificial Intelligence, vol. 1968, Springer, Berlin, 2000, pp. 86–100.
[26] D. Osherson, M. Stob, S. Weinstein, Systems that Learn: An Introduction to Learning Theory for Cognitive and ComputerScientists, MIT Press, Cambridge, 1986.
[27] D. Osherson, S. Weinstein, Criteria of language learning, Information and Control 52 (1982) 123–138.
[28] H. Rogers, Theory of Recursive Functions and Effective Computability, McGraw-Hill, 1967. Reprinted by MIT Press in 1987.
[29] H. Sakamoto, K. Hirata, H. Arimura, Learning elementary formal systems with queries, Theoretical Computer Science A 298 (2003) 21–50.
[30] C. Smith, The power of pluralism for automatic program synthesis, Journal of the ACM 29 (1982) 1144–1165.
[31] T. Zeugmann, S. Lange, A guided tour across the boundaries of learning recursive languages, in: K. Jantke, S. Lange (Eds.), Algorithmic Learning for Knowledge-Based Systems, Lecture Notes in Artificial Intelligence, vol. 961, Springer, Berlin, 1995, pp. 190–258.