Uncountable Automatic Classes and Learning

Sanjay Jain[*], Qinglong Luo[†], Pavel Semukhin[‡] and Frank Stephan[§]

Abstract

In this paper we consider uncountable classes recognizable by ω-automata and investigate suitable learning paradigms for them. In particular, the counterparts of explanatory, vacillatory and behaviourally correct learning are introduced for this setting. Here the learner reads in parallel the data of a text for a language L from the class plus an ω-index α and outputs a sequence of ω-automata such that all but finitely many of these ω-automata accept the index α if and only if α is an index for L.

It is shown that any class is behaviourally correct learnable if and only if it satisfies Angluin's tell-tale condition. For explanatory learning, such a result needs that a suitable indexing of the class is chosen. On the one hand, every class satisfying Angluin's tell-tale condition is vacillatorily learnable in every indexing; on the other hand, there is a fixed class such that the level of the class in the hierarchy of vacillatory learning depends on the indexing of the class chosen.

We also consider a notion of blind learning. On the one hand, a class is blind explanatorily (vacillatorily) learnable if and only if it satisfies Angluin's tell-tale condition and is countable; on the other hand, for behaviourally correct learning, there is no difference between the blind and non-blind version.

This work establishes a bridge between the theory of ω-automata and inductive inference (learning theory).

1 Introduction

Usually, in learning theory one considers classes consisting of countably many languages from some countable domain. A typical example here is the class of all recursive subsets of $\{0, 1, 2\}^*$, where $\{0, 1, 2\}^*$ is the set of all finite strings over the alphabet $\{0, 1, 2\}$. However, each countably infinite domain has uncountably many subsets, and thus we miss out many potential targets when we consider only countable classes. The main goal of this paper is to

[*]Department of Computer Science, National University of Singapore, Singapore 117417, Republic of Singapore, sanjay@comp.nus.edu.sg. Supported in part by NUS grant R252-000-308-112.

[†]Department of Computer Science, National University of Singapore, Singapore 117417, Republic of Singapore, qlluo@singnet.com.sg.

[‡]Department of Mathematics, National University of Singapore, Singapore 119076, Republic of Singapore, pavel@semukhin.name. Supported by NUS grant R146-000-114-112.

[§]Departments of Mathematics and Computer Science, National University of Singapore, Singapore 119076, Republic of Singapore, fstephan@comp.nus.edu.sg. Supported in part by NUS grants R146-000-114-112 and R252-000-308-112.

find a generalization of the classical model of learning which would be suitable for working with uncountable classes of languages. The classes which we consider can be uncountable, but they still have some structure, namely, they are recognizable by Büchi automata. We will investigate, how the classical notions of learnability have to be adjusted in this setting in order to obtain meaningful results. To explain our approach in more detail, we first give an overview of the classical model of inductive inference which is the underlying model of learning in our paper.

Consider a class $\mathcal{L} = \{L_i\}_{i \in I}$, where each language L_i is a subset of Σ^*, the set of finite strings over an alphabet Σ. In a classical model of learning, which was introduced and studied by Gold [9], a learner M receives a sequence of all the strings from a given language $L \in \mathcal{L}$, possibly with repetitions. Such a sequence is called a text for the language. After reading the first n strings from the texts, the learner outputs a hypothesis i_n about what the target language might be. The learner succeeds if it eventually converges to an index that correctly describes the language to be learnt, that is, if $\lim_n i_n = i$ and $L = L_i$. If the learner succeeds on all texts for all languages from a class, then we say that it learns this class. This is the notion of explanatory learning (**Ex**). Such a model became the standard one for the learnability of countable classes. Besides **Ex**, several other paradigms for learning have been considered like, e.g., behaviourally correct (**BC**) learning [3], vacillatory or finite explanatory (**FEx**) learning [8], partial identification (**Part**) [13] and so on.

The indices that the learner outputs are usually finite objects like natural numbers or finite strings. For example, Angluin [1] initiated the research on learnability of uniformly recursive families indexed by natural numbers, and, in recent work, Jain, Luo and Stephan [10] considered automatic indexings by finite strings in place of uniformly recursive indexings. The collection of such finite indices is countable, and hence we can talk only about countable classes of languages. On the other hand, the collection of all the subsets of Σ^* is uncountable, and it looks too restrictive to consider only countable classes. Because of this, it is interesting to find a generalization of the classical model which will allow us to study the learnability of uncountable classes.

Below is an informal description of the learning model that we investigate in this paper. First, since we are going to work with uncountable classes, we need uncountably many indices to index a class to be learnt. For this purpose we will use infinite strings (or ω-strings) over a finite alphabet. Next, we want such indexings to be effective or "computable" in some sense. There are computing machines, called Büchi automata or ω-automata, which can be used naturally for processing ω-strings. They were first introduced by Büchi [6, 7] to prove the decidability of S1S, the monadic second-order theory of the natural numbers with successor function $S(x) = x+1$. Because of this and other decidability results, the theory of ω-automata has become a popular area of research in theoretical computer science (see, e.g., [14]). Taking these points into account, we will assume that a class to be learnt has an indexing by ω-strings which is Büchi recognizable.

The main difference between our model of learning and the classical one is that the learner does not output hypotheses as it processes a text. The reason for this is that it is not possible to output an arbitrary infinite string in a finite amount of time. Instead, in our model, the learner is presented with an index α and a text T, and it must decide whether T is a text for the set with the index α. During its work, the learner outputs an infinite sequence of Büchi automata $\{A_n\}_{n \in \omega}$ (where ω denotes the set of natural numbers) such that A_n accepts the index α if and only if the learner at stage n thinks that T is indeed a text for the set with the index α. The goal of the learner is to converge in the limit to the right answer.

As one can see from the description above, the outputs of a learner take the form of ω-automata instead of just binary answers 'yes' or 'no'. We chose such definition due to the fact that a learner can read only a finite part of an infinite index in a finite amount of time. If we required that a learner outputs its 'yes' or 'no' answer based on such finite information, then our model would become too restrictive. On the other hand, a Büchi automaton allows a learner to encode additional infinitary conditions that have to be verified before the index will be accepted or rejected, for example, if the index contains infinitely many 1's or not. This approach makes a learner more powerful, and more nontrivial classes become learnable.

Probably the most interesting property of our model is that for many learning criteria, the learnability coincides with Angluin's classical tell-tale condition for the countable case (see the table at the end of this section). Angluin's condition states that for every set L from a class $\mathcal{L}$, there is a finite subset $D_L \subseteq L$ such that for any other $L' \in \mathcal{L}$ with $D_L \subseteq L' \subseteq L$ we have that $L' = L$. It is also well-known that in the classical case, every r.e. class is learnable according to the criterion of partial identification [13]. We will show that in our model every ω-automatic class can be learnt according to this criterion.

The results described above suggest that the notions defined in this paper match the intuition of learnability, and that our model is a natural one suitable for investigating the learnability of uncountable classes of languages.

We also consider a notion of blind learning. A learner is called blind if it does not see an index presented to it. Such a learner can see only an input text, but nevertheless it must decide whether the index and the text represent the same language. It turns out that for the criterion of behaviourally correct learning, the blind learners are as powerful as the non-blind ones, but for the other learning criteria this notion becomes more restrictive.

The reader can find all formal definitions of the notions discussed here and some necessary preliminaries in the next section. We summarize our results:

Criterion	Condition	Indexing	Theorems
Ex	ATTC	New	4.1, 5.3
FEx	ATTC	Original	3.1, 5.3
BC	ATTC	Original	5.3
Part	Any class	Original	6.1
BlindBC	ATTC	Original	5.1, 5.3
BlindEx	ATTC & Countable	Original	5.2
BlindFEx	ATTC & Countable	Original	5.2
BlindPart	Countable	Original	6.2

In this table, the first column lists the learning criteria that we studied. Here, **Ex** stands for explanatory learning, **BC** for behaviourally correct learning, **FEx** for finite explanatory or vacillatory learning, and **Part** for partial identification. A prefix **Blind** denotes the blind version of the corresponding criterion. The second column describes equivalent conditions that an automatic class must satisfy for being learnable under the given learning criterion of the first column. Here, ATTC means that the class must satisfy Angluin's tell-tale condition, and Countable means that the class must be countable. The next column indicates whether the learner uses the original indexing of the class or a new one. The last column gives a reference to a theorem/corollary where the result is proved.

2 Preliminaries

An ω-automaton is essentially a finite automaton operating on ω-strings with an infinitary acceptance condition which decides — depending upon the infinitely often visited nodes — which ω-strings are accepted and which are rejected. For a general background on the theory of finite automata the reader is referred to [11].

Definition 2.1 (Büchi [6, 7])**.** A *nondeterministic ω-automaton* is a tuple $A = (S, \Sigma, I, T)$, where

(a) S is a finite set of states,
(b) Σ is a finite alphabet,
(c) $I \subseteq S$ is the set of initial states, and
(d) T is the transition function $T : S \times \Sigma \to \mathcal{P}(S)$, where $\mathcal{P}(S)$ is the power set of S.

An automaton A is *deterministic* if and only if $|I| = 1$, and for all $s \in S$ and $a \in \Sigma$, $|T(s, a)| = 1$.

An *ω-string* over an alphabet Σ is a function $\alpha : \omega \to \Sigma$, where ω is the set of natural numbers. We often identify an ω-string with the infinite sequence $\alpha = \alpha_0\alpha_1\alpha_2\dots$, where $\alpha_i = \alpha(i)$. Let Σ^* and Σ^ω denote the set of all *finite strings* and the set of all *ω-strings* over the alphabet Σ, respectively.

We always assume that the elements of an alphabet Σ are linearly ordered. This order can be extended to the length-lexicographical order $\leq_{llex}$ on Σ^*; here $x \leq_{llex} y$ if and only if $|x| < |y|$ or $|x| = |y| \wedge x \leq_{lex} y$, where $\leq_{lex}$ is the standard lexicographical order.

Given an ω-automaton $A = (S, \Sigma, I, T)$ and an ω-string α, a *run* of A on α is an ω-string

$$r = s_0 \dots s_n s_{n+1} \dots \in S^\omega$$

such that $s_0 \in I$ and for all n, $s_{n+1} \in T(s_n, \alpha_n)$. Note that if an ω-automaton A is deterministic, then for every α, there is a unique run of A on α. In this case we will use the notation $St_A(\alpha, k)$ to denote the state of A after it has read the first k symbols of α.

Definition 2.2. Let $\mathit{Inf}(r)$ denote the *infinity set* of a run r, that is,

$$\mathit{Inf}(r) = \{s \in S : s \text{ appears infinitely often in } r\}.$$

We define the following accepting conditions for the run r:

1) *Büchi condition* is determined by a subset $F \subseteq S$. The run r is accepting if and only if $\mathit{Inf}(r) \cap F \neq \emptyset$.

2) *Muller condition* is determined by a subset $\mathcal{F} \subseteq \mathcal{P}(S)$. The run r is accepting if and only if $\mathit{Inf}(r) \in \mathcal{F}$.

3) *Rabin condition* is determined by $\Omega = \{(F_1, G_1), \ldots, (F_h, G_h)\}$, where all F_i and G_i are subsets of S. The run r is accepting if and only if there is an i such that $1 \leq i \leq h$, $\mathit{Inf}(r) \cap F_i \neq \emptyset$ and $\mathit{Inf}(r) \cap G_i = \emptyset$.

It can be shown that all these acceptance conditions are equivalent in the sense that if a language is accepted by an automaton according to one of the above acceptance criteria, it can also be accepted by an automaton according to any other of these criteria (see [11]). Therefore, we assume that we have fixed one of the acceptance conditions defined above and say that an ω-automaton A *accepts* a string α if and only if there is a run of A on α that satisfies this condition. Let $L(A)$ denote the set of strings accepted by an automaton A according to the chosen acceptance condition.

Furthermore, every ω-automaton is equivalent to a deterministic one with Muller acceptance condition (again, see [11]). Thus, if not explicitly stated otherwise, by an *automaton* we will always mean a *deterministic ω-automaton with Muller acceptance condition.*

Definition 2.3 (Khoussainov and Nerode [11, 12])**.** 1) A *finite automaton* is a tuple $A = (S, \Sigma, I, T, F)$, where S, Σ, I and T are the same as in the definition of an ω-automaton, and $F \subseteq S$ is the set of final states.

2) For a finite string $w = a_0 \ldots a_{n-1} \in \Sigma^*$, a *run* of A on w is a sequence $s_0 \ldots s_n \in S^*$ such that $s_0 \in I$ and $s_{i+1} \in T(s_i, a_i)$ for all $i \leq n-1$. The run is *accepting* if and only if $s_n \in F$. The string $w = a_0 \ldots a_{n-1}$ is *accepted* by A if and only if there is an accepting run of A on w.

Definition 2.4. 1) A *convolution* of k ω-strings $\alpha_1, \ldots, \alpha_k \in \Sigma^\omega$ is an ω-string $\otimes(\alpha_1, \ldots, \alpha_k)$ over the alphabet Σ^k defined as

$$\otimes(\alpha_1, \ldots, \alpha_k)(n) = (\alpha_1(n), \ldots, \alpha_k(n)) \text{ for every } n \in \omega.$$

2) A *convolution* of k finite strings $w_1, \ldots, w_k \in \Sigma^*$ is a string $\otimes(w_1, \ldots, w_k)$ of length $l = \max\{|w_1|, \ldots, |w_k|\}$ over the alphabet $(\Sigma \cup \{\#\})^k$, where $\#$ is a new padding symbol, defined as

$$\otimes(w_1, \ldots, w_k)(n) = (v_1(n), \ldots, v_k(n)) \text{ for every } n < l,$$

where for each $i = 1, \ldots, k$ and $n < l$,

$$v_i(n) = \begin{cases} w_i(n) & \text{if } n < |w_i| \\ \# & \text{otherwise.} \end{cases}$$

3) Correspondingly one defines the convolution of finite strings and ω-strings: one identifies each finite string σ with the ω-string $\sigma\#^\omega$ and forms then the corresponding convolution of ω-strings.

4) A *convolution* of k-ary relation R on finite or ω-strings is defined as

$$\otimes R = \{\otimes(x_1, \ldots, x_k) : (x_1, \ldots, x_k) \in R\}.$$

5) A relation R on finite or ω-strings is *automatic* if and only if its convolution $\otimes R$ is recognizable by a finite or an ω-automaton, respectively.

For the ease of notation, we often just write (x, y) instead of $\otimes(x, y)$ and so on. It is well-known that the automatic relations are closed under union, intersection, projection and complementation. In general, the following theorem holds, which we will often use in this paper.

Theorem 2.5 (Blumensath and Grädel [4, 5])**.** *If a relation R on ω-strings is definable from other automatic relations $R_1, \ldots, R_k$ by a first-order formula, then R itself is automatic.*

Remark 2.6. 1) If we use additional parameters in a first-order definition of a relation R, then these parameters must be ultimately periodic strings.

2) Furthermore, in a definition of a relation R we can use first-order variables of two sorts, namely, one ranging over ω-strings and one ranging over finite strings. We can do this because every finite string v can be identified with its ω-expansion $v\#^\omega$, and the set of all ω-expansions of the finite strings over alphabet Σ is automatic.

A *class* $\mathcal{L}$ is a collection of sets of finite strings over some alphabet Γ, i.e., $\mathcal{L} \subseteq \mathcal{P}(\Gamma^*)$. An *indexing* for a class $\mathcal{L}$ is an onto mapping $f : I \to \mathcal{L}$, where I is the set of indices. We will often denote the indexing as $\{L_\alpha\}_{\alpha \in I}$, where $L_\alpha = f(\alpha)$.

An indexing $\{L_\alpha\}_{\alpha \in I}$ is *automatic* if and only if I is an automatic subset of Σ^ω for some alphabet Σ and the relation $\{(x, \alpha) : x \in L_\alpha\}$ is automatic. A class is *automatic* if and only if it has an automatic indexing. If it is not stated otherwise, *all indexings and all classes considered herein are assumed to be automatic.*

Remark 2.7. According to the definition, an automatic class always comes with an automatic indexing. However we will often say just an "automatic class" instead of an "automatic class with a given automatic indexing." Such abbreviation makes sense because many results of the paper does not depend on the particular choice of an indexing for a class. This can be seen from the table in the introduction section that summarizes the main results. The fact that a learner uses the original indexing of the class actually means that the choice of such indexing is not important. In some cases, where an indexing is important, it will be mentioned explicitly.

Example 2.8. Here are some examples of automatic classes:

1) the class of all open intervals $I = \{q \in \mathbb{D} : p < q < r\}$ of dyadic rationals where the border points p and r can be any real numbers;
2) the class of such intervals where $r - p$ is equal to 1 or 2 or 3;
3) the class of all sets of finite strings which are given as the prefixes of an infinite sequence;
4) the class of all sets of natural numbers in unary coding.

On the other hand, the class of all finite sets of strings over the alphabet $\{0, 1\}$ is not automatic.

A *text* is an ω-string T of the form

$$T = u_0, u_1, u_2, \ldots,$$

such that each u_i is either equal to the pause symbol $\#$ or belongs to Γ^*, where Γ is some alphabet. Note that the comma "$,$" is also part of the text and servers as a delimiter for u_i's. We call u_i the i-th *input* of the text. The *content* of a text T is the set $\mathrm{content}(T) = \{u_i : u_i \neq \#\}$. If $\mathrm{content}(T)$ is equal to a set $L \subseteq \Gamma^*$, then we say that T is a text for L. A *canonical text* for an infinite set L is the listing of all the strings from L in length-lexicographical order. A *canonical text* for a finite set L starts with the listing of all the strings from L in length-lexicographical order, and ends in $\#^\omega$.

Definition 2.9. Let Γ and Σ be alphabets for sets and indices, respectively. A *learner* is a Turing machine $\mathcal{M}$ that has the following:

1) two read-only tapes: one for an ω-string from Σ^ω representing an index and one for a text for a set $L \subseteq \Gamma^*$;
2) one write-only output tape on which $\mathcal{M}$ writes a sequence of automata (in a suitable coding);
3) one read-write working tape.

Let $\mathrm{Ind}(\mathcal{M}, \alpha, T, s)$ and $\mathrm{Txt}(\mathcal{M}, \alpha, T, s)$ denote the number of symbols read in the index and text tapes by learner $\mathcal{M}$ up to step s when it processes an index α and a text T. Without loss of generality, we will assume that

$$\lim_{s\to\infty} \mathrm{Ind}(\mathcal{M}, \alpha, T, s) = \lim_{s\to\infty} \mathrm{Txt}(\mathcal{M}, \alpha, T, s) = \infty$$

for any α and T. By $\mathcal{M}(\alpha, T, k)$ we denote the k-th automaton output by learner $\mathcal{M}$ when processing an index α and a text T. Without loss of generality, for the learning criteria considered in this paper, we assume that $\mathcal{M}(\alpha, T, k)$ is defined for all k.

Definition 2.10 (Based on [3, 8, 9, 13])**.** Let a class $\mathcal{L} = \{L_\alpha\}_{\alpha \in I}$ (together with its indexing) and a learner $\mathcal{M}$ be given. We say that

1) $\mathcal{M}$ **BC**-*learns* $\mathcal{L}$ if and only if for any index $\alpha \in I$ and any text T with $\mathrm{content}(T) \in \mathcal{L}$, there exists n such that for every $m \geq n$,

$$\mathcal{M}(\alpha, T, m) \text{ accepts } \alpha \quad \text{if and only if} \quad L_\alpha = \mathrm{content}(T).$$

2) $\mathcal{M}$ **Ex**-*learns* $\mathcal{L}$ if and only if for any index $\alpha \in I$ and any text T with $\mathrm{content}(T) \in \mathcal{L}$, there exists n such that for every $m \geq n$, $\mathcal{M}(\alpha, T, m) = \mathcal{M}(\alpha, T, n)$ and

$$\mathcal{M}(\alpha, T, m) \text{ accepts } \alpha \quad \text{if and only if} \quad L_\alpha = \mathrm{content}(T).$$

3) $\mathcal{M}$ **FEx**-*learns* $\mathcal{L}$ if and only if $\mathcal{M}$ **BC**-learns $\mathcal{L}$ and for any $\alpha \in I$ and any text T with $\mathrm{content}(T) \in \mathcal{L}$, the set $\{\mathcal{M}(\alpha, T, n) : n \in \omega\}$ is finite.

4) $\mathcal{M}$ $\mathbf{FEx}_k$-*learns* $\mathcal{L}$ if and only if $\mathcal{M}$ **BC**-learns $\mathcal{L}$ and for any $\alpha \in I$ and any text T with $\mathrm{content}(T) \in \mathcal{L}$, there exists n such that

$$|\{\mathcal{M}(\alpha, T, m) : m \geq n\}| \leq k.$$

5) $\mathcal{M}$ **Part**-*learns* $\mathcal{L}$ if and only if for any $\alpha \in I$ and any T with content$(T) \in \mathcal{L}$, there exists a unique automaton A such that for infinitely many m, $\mathcal{M}(\alpha, T, m) = A$, and for this unique A,

$$A \text{ accepts } \alpha \quad \text{if and only if} \quad L_\alpha = \text{content}(T).$$

Here the abbreviations **BC**, **Ex**, **FEx** and **Part** stand for 'behaviourally correct', 'explanatory', 'finite explanatory' and 'partial identification', respectively; 'finite explanatory learning' is also called 'vacillatory learning'. We will also use the notations **BC**, **Ex**, **FEx**, $\mathbf{FEx}_k$ and **Part** to denote the collection of classes (with corresponding indexings) that are **BC**-, **Ex**-, **FEx**-, $\mathbf{FEx}_k$- and **Part**-learnable, respectively.

Definition 2.11. A learner is called *blind* if it does not see the tape which contains an index. The classes that are blind **BC**-, **Ex**-, etc. learnable are denoted as **BlindBC**, **BlindEx**, etc., respectively.

Definition 2.12 (Angluin [1])**.** We say that a class $\mathcal{L}$ satisfies *Angluin's tell-tale condition* if and only if for every $L \in \mathcal{L}$ there is a finite $D_L \subseteq L$ such that for every $L' \in \mathcal{L}$, if $D_L \subseteq L' \subseteq L$ then $L' = L$. Such D_L is called a *tell-tale* set for L.

Using techniques similar to those that were introduced in [1], it can be shown that

Fact 2.13. *If a class $\mathcal{L}$ is* **BC***-learnable, then $\mathcal{L}$ satisfies Angluin's tell-tale condition.*

The converse will also be shown to be true, hence for automatic classes one can equate "$\mathcal{L}$ is learnable" with "$\mathcal{L}$ satisfies Angluin's tell-tale condition". Note that the second and the third class given in Example 2.8 satisfy Angluin's tell-tale condition.

3 Vacillatory Learning

In the following it is shown that every learnable class can even be vacillatorily learnt and that the corresponding **FEx**-learner uses overall on all possible inputs only a fixed number of automata.

Theorem 3.1. *Let $\{L_\alpha\}_{\alpha \in I}$ be a class that satisfies Angluin's tell-tale condition. Then there are finitely many automata $A_1, \ldots, A_c$ and an* **FEx***-learner $\mathcal{M}$ for the class $\{L_\alpha\}_{\alpha\in I}$ with the property that for any $\alpha \in I$ and any text T for a set from $\{L_\alpha\}_{\alpha\in I}$, the learner $\mathcal{M}$ oscillates only between some of the automata $A_1, \ldots, A_c$ on α and T.*

Proof. Let M be a deterministic automaton recognizing the relation $\{(x, \alpha) : x \in L_\alpha\}$, and let N be a deterministic automaton recognizing

$$\{\, (x, \alpha) \ : \ \{y \in L_\alpha : y \leq_{llex} x\} \text{ is a tell-tale for } L_\alpha \,\}.$$

Such an N exists since the relation is first-order definable from '$x \in L_\alpha$' and $\leq_{llex}$ by the formula:

$$N \text{ accepts } (x,\alpha) \quad \Longleftrightarrow \quad \forall \alpha' \in I \, \Big(\text{if } \ \forall y\,((y \in L_\alpha \ \&\ y \leq_{llex} x) \ \to \ y \in L_{\alpha'}) \ \& \\ \forall y\,(y \in L_{\alpha'} \to y \in L_\alpha), \text{ then } \forall y\,(y \in L_{\alpha'} \leftrightarrow y \in L_\alpha)\Big).$$

For each $\alpha \in I$, consider an equivalence relation $\equiv_{M,\alpha}$ defined as

$$x \equiv_{M,\alpha} y \iff \begin{array}{l}\text{there is a } t > \max\{|x|,|y|\} \text{ such that} \\ St_M(\otimes(x,\alpha),t) = St_M(\otimes(y,\alpha),t).\end{array}$$

An equivalence relation $\equiv_{N,\alpha}$ is defined in a similar way.

Note that the number of the equivalence classes of $\equiv_{M,\alpha}$ and $\equiv_{N,\alpha}$ are bounded by the number of states of M and N, respectively. Also, for every x, y, if $x \equiv_{M,\alpha} y$ then $x \in L_\alpha \leftrightarrow y \in L_\alpha$. Therefore, L_α is the union of finitely many equivalence classes of $\equiv_{M,\alpha}$.

Let m and n be the number of states of M and N, respectively. Consider the set of all finite tables $U = \{U_{i,j} : 1 \le i \le m,\ 1 \le j \le n\}$ of size $m \times n$ such that each $U_{i,j}$ is either equal to a subset of $\{1,\dots,i\}$ or to a special symbol *Reject*. Note that the number of such tables is finite.

With each such table U we will associate an automaton A as described below. The algorithm for learning $\{L_\alpha\}_{\alpha\in I}$ is now roughly as follows. On every step, the learner $\mathcal{M}$ reads a finite part of the input text and the index and based on this information constructs a table U. After that $\mathcal{M}$ outputs the automaton associated with U.

First, we describe the construction of an automaton A for each table U. For every $\alpha \in I$, let $m(\alpha)$ and $n(\alpha)$ be the numbers of the equivalence classes of $\equiv_{M,\alpha}$ and $\equiv_{N,\alpha}$, respectively. Also, let

$$x_1 <_{llex} \cdots <_{llex} x_{m(\alpha)}$$

be the length-lexicographically least representatives of the equivalence classes of $\equiv_{M,\alpha}$. Our goal is to construct A such that

$$A \text{ accepts } \alpha \iff \begin{array}{l}U_{m(\alpha),n(\alpha)} \text{ is a subset of } \{1,\dots,m(\alpha)\}, \text{ and } L_\alpha \text{ is equal} \\ \text{to the union of the } \equiv_{M,\alpha}\text{-equivalence classes that are} \\ \text{represented by } x_i\text{'s with } i \in U_{m(\alpha),n(\alpha)}.\end{array}$$

Let $EqSt_M(\alpha,x,y,z)$ be the relation defined as

$$EqSt_M(\alpha,x,y,z) \iff St_M(\otimes(x,\alpha),|z|) = St_M(\otimes(y,\alpha),|z|).$$

The relation $EqSt_N(\alpha,x,y,z)$ is defined similarly. Note that these relations are automatic.

Instead of constructing A explicitly, we will show that the language which A needs to recognize is first-order definable from $EqSt_M(\alpha,x,y,z)$, $EqSt_N(\alpha,x,y,z)$ and the relations recognized by M and N.

First, note that the equivalence relation $x \equiv_{M,\alpha} y$ can be defined by a formula:

$$\exists z\ (|z| > \max\{|x|,|y|\} \text{ and } EqSt_M(\alpha,x,y,z)).$$

Similarly one can define $x \equiv_{N,\alpha} y$. The fact that $\equiv_{M,\alpha}$ has exactly k many equivalence classes can be expressed by a formula:

$$ClNum_{M,k}(\alpha) = \exists x_1 \dots \exists x_k \Big(\bigwedge_{1\le i<j\le k} x_i \not\equiv_{M,\alpha} x_j \ \ \&\ \ \forall y \bigvee_{1\le i\le k} y \equiv_{M,\alpha} x_i \Big).$$

Again, $\mathit{ClNum}_{N,k}(\alpha)$ expresses the same fact for $\equiv_{N,\alpha}$. Finally, the fact that A accepts α can be expressed by the following first-order formula:

$$\bigvee_{(i,j)\ :\ U_{i,j}\neq \mathit{Reject}} \Big(\mathit{ClNum}_{M,i}(\alpha)\ \&\ \mathit{ClNum}_{N,j}(\alpha)\ \&\ \exists x_1 \ldots \exists x_i \ \big(x_1 <_{llex} \cdots <_{llex} x_i\ \& \\ \forall z \, (z \in L_\alpha \leftrightarrow \bigvee_{k\in U_{i,j}} z \equiv_{M,\alpha} x_k)\ \ \&\ \bigwedge_{1\le k\le i} \forall y \, (y <_{llex} x_k \to y \not\equiv_{M,\alpha} x_k) \big) \Big).$$

We now describe the algorithm for learning the class $\{L_\alpha\}_{\alpha\in I}$. We will use the notation $x \equiv_{M,\alpha,s} y$ as an abbreviation of

"there is t such that $s \ge t > \max\{|x|, |y|\}$ and $St_M(\otimes(x,\alpha),t) = St_M(\otimes(y,\alpha),t)$."

As before, let m and n be the numbers of states of automata M and N, respectively. At step s, $\mathcal{M}$ computes $\le_{llex}$ least representatives of the equivalence classes of $\equiv_{M,\alpha,s}$ and $\equiv_{N,\alpha,s}$ on the strings with length shorter than s. In other words, it computes $x_1,\ldots,x_p$ and $y_1,\ldots,y_q$ such that

a) x_1 is the empty string,
b) x_{k+1} is the $\le_{llex}$ least $x >_{llex} x_k$ such that $|x| \le s$ and $x \not\equiv_{M,\alpha,s} x_i$ for all $i \le k$. If such x does not exists then the process stops.

The sequence $y_1,\ldots,y_q$ is computed in a similar way.

Next, $\mathcal{M}$ constructs a table U of size $m\times n$. For every i and j, the value of $U_{i,j}$ is defined as follows. If $i > p$ or $j > q$, then let $U_{i,j} = \mathit{Reject}$. Otherwise, let τ_s be the initial segment of the input text T consisting of the first s strings in the text T. Check if the following two conditions are satisfied:

1) for every $x, x' \le_{llex} y_j$, if $x \equiv_{M,\alpha,s} x'$, then $x \in \mathrm{content}(\tau_s)$ if and only if $x' \in \mathrm{content}(\tau_s)$,
2) for every $k \le i$ and every y, if $y \in \mathrm{content}(\tau_s)$ and $y \equiv_{M,\alpha,s} x_k$, then $x_k \in \mathrm{content}(\tau_s)$.

If yes, then let $U_{i,j} = \{k :\ k \le i \text{ and } x_k \in \mathrm{content}(\tau_s)\}$. Otherwise, let $U_{i,j} = \mathit{Reject}$. After U is constructed, $\mathcal{M}$ outputs an automaton A associated with U as described above. As the number of different possible U is finite, the number of distinct corresponding automata output by $\mathcal{M}$ is finite.

Recall that $\mathcal{M}(\alpha,T,s)$ is the automaton output by learner $\mathcal{M}$ at step s when processing the index α and the text T. To prove that the algorithm is correct we need to show that for every $\alpha \in I$ and every text T such that $\mathrm{content}(T) \in \{L_\beta\}_{\beta\in I}$,

a) if $\mathrm{content}(T) = L_\alpha$ then for almost all s, $\mathcal{M}(\alpha,T,s)$ accepts α,
b) if $\mathrm{content}(T) \neq L_\alpha$ then for almost all s, $\mathcal{M}(\alpha,T,s)$ rejects α.

Recall that $m(\alpha)$ and $n(\alpha)$ are the numbers of the equivalence classes of $\equiv_{M,\alpha}$ and $\equiv_{N,\alpha}$, respectively. Note that there is a step s_0 after which the values $x_1 <_{llex} \cdots <_{llex} x_{m(\alpha)}$ and $y_1 <_{llex} \cdots <_{llex} y_{n(\alpha)}$ computed by $\mathcal{M}$ will always be equal to the $\le_{llex}$ least representatives of the equivalence classes of $\equiv_{M,\alpha}$ and $\equiv_{N,\alpha}$, respectively.

Suppose that $\mathrm{content}(T) = L_\alpha$. Hence, there is $s_1 \ge s_0$ such that for every $s \ge s_1$ the following conditions are satisfied:

1) for every $k \leq m(\alpha)$, $x_k \in \mathrm{content}(\tau_s)$ if and only if $x_k \in \mathrm{content}(T)$,
2) for every $x, x' \leq_{llex} y_{n(\alpha)}$, if $x \equiv_{M,\alpha,s} x'$, then $x \in \mathrm{content}(\tau_s)$ if and only if $x' \in \mathrm{content}(\tau_s)$,
3) for every $k \leq m(\alpha)$ and every y, if $y \in \mathrm{content}(\tau_s)$ and $y \equiv_{M,\alpha,s} x_k$, then $x_k \in \mathrm{content}(\tau_s)$.

The last two conditions are satisfied since $\mathrm{content}(T) = L_\alpha$ is the union of finitely many $\equiv_{M,\alpha}$ equivalence classes. Therefore, on every step $s \geq s_1$, the learner $\mathcal{M}$ constructs a table U such that $U_{m(\alpha),n(\alpha)} = \{k : k \leq m(\alpha) \text{ and } x_k \in \mathrm{content}(T)\}$. By our construction of the automaton A associated with U, A accepts α if $L_\alpha = \{y : y \equiv_{M,\alpha} x_k \text{ for some } x_k \in \mathrm{content}(T)\}$. But since $\mathrm{content}(T) = L_\alpha$, this condition is satisfied.

Now suppose that $\mathrm{content}(T) \neq L_\alpha$. Note that for every $s \geq s_0$, $y_{n(\alpha)}$ computed by $\mathcal{M}$ at step s has the property that $D_\alpha = \{x \in L_\alpha : x \leq_{llex} y_{n(\alpha)}\}$ is a tell-tale set for L_α. This follows from the definition of the automaton N and the fact that $y_{n(\alpha)}$ is the $\leq_{llex}$ largest among the representatives of the $\equiv_{N,\alpha}$ equivalence classes.

First, consider the case when $D_\alpha \not\subseteq \mathrm{content}(T)$, that is, there is $x \in L_\alpha$, $x \leq_{llex} y_{n(\alpha)}$ but $x \notin \mathrm{content}(T)$. Let $s_1 \geq s_0$ be such that $x \equiv_{M,\alpha,s_1} x_k$ for some $k \leq m(\alpha)$. Note that $x_k \leq_{llex} x$ since x_k is the minimal representative in its equivalence class. If for some $s_2 \geq s_1$, $x_k \in \mathrm{content}(\tau_{s_2})$, then from this step on $U_{m(\alpha),n(\alpha)}$ will be equal to *Reject* because of the Condition 1) in the definition of $U_{i,j}$. Hence $\mathcal{M}(\alpha, T, s)$ will reject α for all $s \geq s_2$. If $x_k \notin \mathrm{content}(T)$, then for all $s \geq s_1$, $\mathcal{M}(\alpha, T, s)$ will reject α either due to the fact that $U_{m(\alpha),n(\alpha)} = \textit{Reject}$ at step s, or because $k \notin U_{m(\alpha),n(\alpha)}$ while it should be in $U_{m(\alpha),n(\alpha)}$ since both x and x_k are in L_α.

Now suppose that $D_\alpha \subseteq \mathrm{content}(T)$. Since D_α is a tell-tale set for L_α and $\mathrm{content}(T) \neq L_\alpha$, there is $x \in \mathrm{content}(T) \setminus L_\alpha$. Let $s_1 \geq s_0$ be such that $x \in \mathrm{content}(\tau_{s_1})$ and $x \equiv_{M,\alpha,s_1} x_k$ for some $k \leq m(\alpha)$. If $x_k \notin \mathrm{content}(T)$ then for every $s \geq s_1$, $U_{m(\alpha),n(\alpha)} = \textit{Reject}$ and $\mathcal{M}(\alpha, T, s)$ will reject α. If there is $s_2 \geq s_1$ such that $x_k \in \mathrm{content}(\tau_{s_2})$, then for every $s \geq s_2$ either $U_{m(\alpha),n(\alpha)} = \textit{Reject}$ or $k \in U_{m(\alpha),n(\alpha)}$. In both cases $\mathcal{M}(\alpha, T, s)$ will reject α since $x_k \notin L_\alpha$. □

Definition 3.2. 1) Let $\alpha \in \{0, 1, \ldots, k\}^\omega$ and $\beta \in \{1, \ldots, k\}^\omega$. The function $f_{\alpha,\beta}$ is defined as follows:

$$f_{\alpha,\beta}(n) = \begin{cases} \alpha(m) & \text{if } m = \min\{x \geq n : \alpha(x) \neq 0\}, \\ \limsup_{x \to \infty} \beta(x) & \text{if such } m \text{ does not exist.} \end{cases}$$

Let $L_{\alpha,\beta}$ be the set of all nonempty finite prefixes of $f_{\alpha,\beta}$, that is,

$$L_{\alpha,\beta} = \{f_{\alpha,\beta}(0) \ldots f_{\alpha,\beta}(n) \ : \ n \in \omega\}.$$

2) Define the class $\mathcal{L}^k$ together with its indexing as follows. Let the index set be

$$J_k = \{(\alpha, \beta) : \alpha \in \{0, 1, \ldots, k\}^\omega, \ \beta \in \{1, 2, \ldots, k\}^\omega\}$$

and let

$$\mathcal{L}^k = \{L_{\alpha,\beta}\}_{(\alpha,\beta) \in J_k}.$$

Note that the class $\mathcal{L}^k$ is uncountable and automatic.

Theorem 3.3. *For every $k \geq 2$, the class $\mathcal{L}^k = \{L_{\alpha,\beta}\}_{(\alpha,\beta)\in J_k}$ is in $\mathbf{FEx}_k \setminus \mathbf{FEx}_{k-1}$.*

Proof. We first show that $\mathcal{L}^k$ is $\mathbf{FEx}_k$-learnable. Let $A_0, A_1, \ldots, A_k$ be automata such that A_0 rejects all ω-strings, and for $i = 1, \ldots, k$

$$A_i \text{ accepts } (\alpha, \beta) \iff \limsup_{x\to\infty} \alpha(x) \neq 0 \text{ or } \limsup_{x\to\infty} \beta(x) = i.$$

A learner $\mathcal{M}$ that $\mathbf{FEx}_k$-learns $\mathcal{L}^k$ acts as follows. At every step s, $\mathcal{M}$ reads the first s inputs from the input text. If all these inputs are equal to #, then $\mathcal{M}$ outputs A_0. Otherwise, let t_s be the longest string among them. Next, $\mathcal{M}$ checks if t_s is consistent with α, that is, if there is a j with $1 \leq j \leq k$ such that for every $n < |t_s|$,

$$t_s(n) = \begin{cases} \alpha(m) & \text{if } m = \min\{x : n \leq x < |t_s| \text{ and } \alpha(x) \neq 0\}, \\ j & \text{if such } m \text{ does not exist.} \end{cases}$$

If t_s is inconsistent with α, then $\mathcal{M}$ outputs only the automaton A_0 from step s onward. Otherwise, in the end of step s the learner $\mathcal{M}$ outputs A_i, where i is the last symbol of t_s. Now it is not hard to verify that this algorithm is correct.

To show that $\mathcal{L}^k$ is not in $\mathbf{FEx}_{k-1}$, assume, for the sake of contradiction, that there is a learner $\mathcal{M}$ that can $\mathbf{FEx}_{k-1}$-learn $\mathcal{L}^k$. First, we need the following two lemmas.

Lemma 3.4. *There are finite strings α', β' and $k-1$ automata $A_1, \ldots, A_{k-1}$ such that*

a) $\alpha' \in \{0, 1, \ldots, k\}^*$, $\beta' \in \{1, \ldots, k\}^*$ and $|\alpha'| = |\beta'|$,

b) for every ω-string β such that $\beta' \subset \beta \in \{1, \ldots, k\}^\omega$, there is a text T for $L_{\alpha'0^\omega,\beta}$ (which can be chosen to be the canonical text for $L_{\alpha'0^\omega,\beta}$) such that the learner $\mathcal{M}$ on index $(\alpha'0^\omega, \beta)$ and text T oscillates only between $A_1, \ldots, A_{k-1}$ after it has seen (α', β').

Proof of Lemma 3.4. Suppose that there are no such α', β' and $A_1, \ldots, A_{k-1}$. In other words, for any α', β' for which property a) holds and any $k-1$ automata $A_1, \ldots, A_{k-1}$, there are an ω-string β with $\beta' \subset \beta \in \{1, \ldots, k\}^\omega$ and an automaton $A \notin \{A_1, \ldots, A_{k-1}\}$ such that $\mathcal{M}$ outputs A *above* (α', β') when processing the index $(\alpha'0^\omega, \beta)$ and the canonical text for $L_{\alpha'0^\omega,\beta}$, that is, $\mathcal{M}$ outputs A at some step after the first step at which it has seen (α', β').

We now show that $\mathcal{L}^k \notin \mathbf{FEx}_{k-1}$ by constructing ω-strings α, β and a text T for $L_{\alpha,\beta}$ such that $\mathcal{M}$ oscillates between more than $k-1$ many automata when processing (α, β) and T. At each step i, we will construct finite strings α_i, α'_i, β_i, β'_i and a finite text segment τ_i such that the following properties hold:

1) $\alpha_i, \alpha'_i \in \{0, 1, \ldots, k\}^*$ and $\beta_i, \beta'_i \in \{1, \ldots, k\}^*$.
2) $|\alpha_i| = |\beta_i|$ and $|\alpha'_i| = |\beta'_i|$.
3) $\alpha'_i \subseteq \alpha_i \subseteq \alpha'_{i+1}$, $\beta'_i \subseteq \beta_i \subseteq \beta'_{i+1}$, and $\tau_i \subseteq \tau_{i+1}$.
4) $\alpha = \bigcup_{i\in\omega} \alpha_i$, $\beta = \bigcup_{i\in\omega} \beta_i$ and $T = \bigcup_{i\in\omega} \tau_i$.
5) For $i > 0$, α_i does not end in 0.
6) τ_i is a finite prefix of the canonical text for $L_{\alpha_i 0^\omega,\beta}$ that contains strings of length not greater than $|\alpha_i|$ (for some β, but since α_i does not end in 0 and since we only consider strings which are shorter than α_i, this β is irrelevant).

7) When the learner $\mathcal{M}$ processes the input (α_i, β_i) and the text segment τ_i, it does not try to read beyond τ_i in the text before it reads beyond the prefix (α'_i, β'_i) of (α_i, β_i).

At step 0, let all α'_0, α_0, β'_0, β_0 and τ_0 be equal to the empty string. At step $i+1$, let $A_1, \ldots, A_{k-1}$ be the last $k-1$ different automata output by $\mathcal{M}$ up to the first step at which it has seen (α'_i, β'_i) when processing (α_i, β_i) on text segment τ_i (if there are less then $k-1$ such automata, then consider the set of all these automata instead of $A_1, \ldots, A_{k-1}$).

By our assumption, there are an ω-string β with $\beta_i \subset \beta \in \{1, \ldots, k\}^\omega$ and an automaton $A \notin \{A_1, \ldots, A_{k-1}\}$ such that $\mathcal{M}$ outputs A above (α_i, β_i) when processing $(\alpha_i 0^\omega, \beta)$ and the canonical text T' for $L_{\alpha_i 0^\omega, \beta}$. Due to property 6), T' extends τ_i.

Now wait until the learner $\mathcal{M}$ outputs $A \notin \{A_1, \ldots, A_{k-1}\}$ on $(\alpha_i 0^\omega, \beta)$ and T' above (α_i, β_i). Let $(\alpha'_{i+1}, \beta'_{i+1})$ and τ_{i+1} be the finite segments of the index and the text seen by that time. Here, if τ_{i+1} does not properly extend τ_i, then we take τ_{i+1} to be the extension of τ_i by one more symbol; furthermore, if τ_{i+1} ends in a middle of a string from T', then we extend τ_{i+1} up to the beginning of the next string.

Let t be the maximum of $|\alpha'_{i+1}| + 1$ and the length of the longest string from τ_{i+1}. Let $\alpha_{i+1} = \alpha'_{i+1} 0^s m$, where $m = \limsup_{x \to \infty} \beta(x)$ and s is chosen in such a way that $|\alpha_{i+1}| = t$. Finally, let β_{i+1} be the prefix of β of length t. This concludes the description of step $i+1$.

Now, by the construction, T is a text for $L_{\alpha,\beta}$ (in fact, the canonical one), and $\mathcal{M}$ oscillates between more then $k-1$ many automata when processing (α, β) and T.

This completes the proof of Lemma 3.4.

Lemma 3.5. *Suppose that there is l such that $1 < l < k$, and there are l many automata $A_1, \ldots, A_l$ together with finite strings α', β' with the following properties:*

a) $\alpha' \in \{0, 1, \ldots, k\}^*$, $\beta' \in \{1, \ldots, k\}^*$ and $|\alpha'| = |\beta'|$,

b) for every ω-string $\beta \supset \beta'$ such that $1 \le \beta(x) \le l+1$ for all $x \ge |\beta'|$, the learner $\mathcal{M}$ on index $(\alpha' 0^\omega, \beta)$ and the canonical text for $L_{\alpha' 0^\omega, \beta}$ oscillates only between $A_1, \ldots, A_l$ after it has seen (α', β').

Then there are $l-1$ many automata $\{A'_1, \ldots, A'_{l-1}\} \subset \{A_1, \ldots, A_l\}$ and finite strings α'', β'' such that

1) $\alpha'' \in \alpha'\{0\}^*$, $\beta'' \in \beta'\{1, \ldots, l+1\}^*$ and $|\alpha''| = |\beta''|$,

2) for every ω-string $\beta \supset \beta''$ such that $1 \le \beta(x) \le l$ for all $x \ge |\beta''|$, the learner $\mathcal{M}$ on index $(\alpha'' 0^\omega, \beta)$ and the canonical text for $L_{\alpha'' 0^\omega, \beta}$ oscillates only between $A'_1, \ldots, A'_{l-1}$ after it has seen (α'', β'').

Proof of Lemma 3.5. Assume that there are no such α'', β'' and $A'_1, \ldots, A'_{l-1}$. Thus, for any $A \in \{A_1, \ldots, A_l\}$ and any α'', β'' for which property 1) holds, there is an ω-string $\beta \in \beta''\{1, \ldots, l\}^\omega$ such that the learner $\mathcal{M}$ outputs A on $(\alpha'' 0^\omega, \beta)$ and the canonical text for $L_{\alpha'' 0^\omega, \beta}$ above (α'', β'').

For n with $1 \le n \le l$, let T_n be the canonical text for $L_{\alpha' 0^\omega, n^\omega}$. We will construct an ω-string $\beta \in \beta'\{1, \ldots, l+1\}^\omega$ with $\limsup_{x \to \infty} \beta(x) = l+1$. Moreover, for every $A \in \{A_1, \ldots, A_l\}$, there will be $n \in \{1, \ldots, l\}$ such that $\mathcal{M}$ outputs A infinitely often on index $(\alpha' 0^\omega, \beta)$ and text T_n. At each step i we will construct a finite string $\beta_i \in \beta'\{1, \ldots, l+1\}^*$ such that $\beta_i \subseteq \beta_{i+1}$ and $\beta = \bigcup_i \beta_i$.

At step 0, let $\beta_0 = \beta'$. At step $i+1$, let $m \in \{1, \dots, l\}$ be such that $m \equiv i+1 \pmod{l}$. By our assumption, there exists an ω-string $\beta \in \beta_i\{1, \dots, l\}^\omega$, and $\mathcal{M}$ outputs A_m on $(\alpha' 0^\omega, \beta)$ and T_n above $(\alpha' 0^s, \beta_i)$, where $n = \limsup_{x \to \infty} \beta(x)$ and $s = |\beta_i| - |\alpha'|$. Now let $\beta'_i \supseteq \beta_i$ be the finite prefix of β seen by $\mathcal{M}$ when it outputs A_m for the first time above $(\alpha' 0^s, \beta_i)$ on text T_n, and let β_{i+1} be equal to $\beta'_i(l+1)$, that is, β'_i followed by number $l+1$. This concludes step $i+1$.

By the construction, $\limsup_{x \to \infty} \beta(x) = l+1$ and for every $m = 1, \dots, l$ and every $r \in \omega$, there is $n \in \{1, \dots, l\}$ such that $\mathcal{M}$ outputs A_m after reading $(\alpha' 0^s, \beta_{r \cdot l + m})$ on text T_n, where $s = |\beta_{r \cdot l + m}| - |\alpha'|$. Therefore, for every $A \in \{A_1, \dots, A_l\}$, there is $n \in \{1, \dots, l\}$ such that $\mathcal{M}$ outputs A infinitely often on $(\alpha' 0^\omega, \beta)$ and T_n.

Since $\limsup_{x \to \infty} \beta(x) = l+1$, each T_n is different from $L_{\alpha' 0^\omega, \beta}$. So, every $A \in \{A_1, \dots, A_l\}$ must reject $(\alpha' 0^\omega, \beta)$. On the other hand, since $\beta \in \beta'\{1, \dots, l+1\}^\omega$, the learner $\mathcal{M}$ on index $(\alpha' 0^\omega, \beta)$ and the canonical text for $L_{\alpha' 0^\omega, \beta}$ oscillates only between $A_1, \dots, A_l$ after it has seen (α', β'). So, there is $A \in \{A_1, \dots, A_l\}$ which is output by $\mathcal{M}$ infinitely often, and this A must accept $(\alpha' 0^\omega, \beta)$. But we just showed that every such A must reject $(\alpha' 0^\omega, \beta)$. This contradiction proves the lemma.

This completes the proof of Lemma 3.5.

By Lemma 3.4, the assumption of Lemma 3.5 holds for $l = k-1$. Now, applying Lemma 3.5 inductively for l from $k-1$ down to 2, we eventually obtain that there are finite strings α', β' and an automaton A such that

1) $\alpha' \in \{0, 1, \dots, k\}^*$, $\beta' \in \{1, \dots, k\}^*$ and $|\alpha'| = |\beta'|$,

2) for every ω-string $\beta \supset \beta'$ such that $\beta(x) \in \{1, 2\}$ for all $x \geq |\beta'|$, the learner $\mathcal{M}$ on index $(\alpha' 0^\omega, \beta)$ and the canonical text for $L_{\alpha' 0^\omega, \beta}$ outputs only A above (α', β').

For $n \in \{1, 2\}$, let T_n be the canonical text for $L_{\alpha' 0^\omega, n^\omega}$. We now construct an ω-string $\beta \in \beta'\{1, 2\}^\omega$ with $\limsup_{x \to \infty} \beta(x) = 2$ such that $\mathcal{M}$ outputs only A on $(\alpha' 0^\omega, \beta)$ and T_1 above (α', β'). Again, for every i, we will construct $\beta_i \in \beta'\{1, 2\}^*$ such that $\beta_i \subseteq \beta_{i+1}$ and $\beta = \bigcup_i \beta_i$.

Let $\beta_0 = \beta'$. Suppose we have constructed β_i. By our assumption, the learner $\mathcal{M}$ on $(\alpha' 0^\omega, \beta_i 1^\omega)$ and T_1 outputs only A above (α', β'). Now wait until the first step when $\mathcal{M}$ outputs A above $(\alpha' 0^s, \beta_i)$, where $s = |\beta_i| - |\alpha'|$. Let $\beta'_i \supseteq \beta_i$ be the finite prefix of $\beta_i 1^\omega$ seen by $\mathcal{M}$ by that time, and let $\beta_{i+1} = \beta'_i 2$.

Since $\limsup_{x \to \infty} \beta(x) = 2$ and since $\mathcal{M}$ outputs only A on $(\alpha' 0^\omega, \beta)$ and T_1 above (α', β'), A must reject $(\alpha' 0^\omega, \beta)$. On the other hand, $\mathcal{M}$ on $(\alpha' 0^\omega, \beta)$ and T_2 outputs only A above (α', β'). Therefore, A must accept $(\alpha' 0^\omega, \beta)$. This contradiction proves the theorem. $\square$

Remark 3.6. The last result can be strengthened in the following sense: for every $k \geq 1$ there is an indexing $\{L_\beta\}_{\beta \in I}$ of the class $\mathcal{L} = \{\{\alpha_0 \alpha_1 \alpha_2 \dots \alpha_{n-1} : n \in \omega\} : \alpha \in \{1, 2\}^\omega\}$ such that $\{L_\beta\}_{\beta \in I}$ is $\mathbf{FEx}_{k+1}$-learnable but not $\mathbf{FEx}_k$-learnable. That is, the class can be kept fixed and only the indexing has to be adjusted. In order to keep the proof above more readable, this adjustment was not implemented there.

4 Explanatory Learning

The main result of this section is that for every class that satisfies Angluin's tell-tale condition, there is an indexing in which the class is explanatorily learnable. A learner which we construct will, in general, have a mind change sequence which is a subsequence of *Reject–Accept–Reject*. That is, at first, the learner may think that the index is wrong and reject it. Then it may change its mind and start to think that the index is correct and accept it. In the end, the learner may change its mind again, and in this case it will keep rejecting the index forever. The mind changes mentioned above are the worst case, and in some situations, such mind changes may not happen. In the second part of this section we will consider such patterns of mind changes in more detail and characterize the classes that can be learnt that way.

Theorem 4.1. *If a class $\mathcal{L} = \{L_\alpha\}_{\alpha\in I}$ satisfies Angluin's tell-tale condition, then there is an indexing for $\mathcal{L}$ such that $\mathcal{L}$ with this indexing is* **Ex***-learnable.*

Proof. Let M be a deterministic automaton recognizing $\{(x,\alpha) : x \in L_\alpha\}$, and Q_M be its set of states. The set J of new indices for $\mathcal{L}$ will consist of convolutions $\otimes(\alpha,\beta,\gamma)$, where $\alpha \in I$, $\beta \in \{0,1\}^\omega$ determines a tell-tale set for L_α, and $\gamma \in \{\mathcal{P}(Q_M)\}^\omega$ keeps track of states of M when it reads $\otimes(x,\alpha)$ for some finite strings $x \in L_\alpha$. To simplify the notations we will write (α,β,γ) instead of $\otimes(\alpha,\beta,\gamma)$. Formally, J is defined as follows:

$$\begin{aligned}(\alpha,\beta,\gamma)\in J \quad \iff \quad & \alpha\in I,\ \beta = 0^n1^\omega \text{ for the minimal } n \text{ such that}\\ & \{x\in L_\alpha : |x|<n\} \text{ is a tell-tale set for } L_\alpha, \text{ and for every } k\\ & \gamma(k) = \{q\in Q_M :\ \exists x\in L_\alpha\,(|x|\le k \text{ and } St_M(\otimes(x,\alpha),k)=q)\}.\end{aligned}$$

We want to show that J is automatic. Again, it is enough to show that it is first-order definable from other automatic relations. We can define β as the lexicographically largest β' that satisfies the formula:

$$\beta'\in 0^*1^\omega \ \ \&\ \ \forall\sigma\in 0^*\,\big((\sigma\subseteq\beta' \ \&\ \sigma0\not\subseteq\beta')\to\{x\in L_\alpha : |x|<|\sigma|\} \text{ is a tell-tale set for } L_\alpha\big).$$

The first-order definition for a tell-tale set is given in the beginning of the proof of Theorem 3.1. All other relations in this definition are clearly automatic.

The definition for γ can be written as

$$\forall\sigma\in 0^* \bigwedge_{q\in Q_M}\Big(q\in\gamma(|\sigma|)\ \leftrightarrow\ \exists x\in L_\alpha\,(\,|x|\le|\sigma|\ \&\ St_M(\otimes(x,\alpha),|\sigma|)=q)\Big).$$

For every $q \in Q_M$, there are automata A_q and B_q that recognize the relations

$$\{(\sigma,\gamma) : \sigma\in 0^*\ \&\ q\in\gamma(|\sigma|)\,\} \quad \text{and} \quad \{(\sigma,x,\alpha) : \sigma\in 0^*\ \&\ St_M(\otimes(x,\alpha),|\sigma|)=q\,\}.$$

Therefore, J is first-order definable from automatic relations, and hence itself is automatic.

We now show that for every finite string x,

$$x\in L_\alpha \iff St_M(\otimes(x,\alpha),|x|)\in\gamma(|x|),$$

provided that γ is correctly defined from α as in the definition of J. Indeed, if $x \in L_\alpha$, then $St_M(\otimes(x,\alpha),|x|) \in \gamma(|x|)$ by the definition of γ. On the other hand, if $St_M(\otimes(x,\alpha),|x|) \in \gamma(|x|)$, then, again by the definition of γ, there is $y \in L_\alpha$ with $|y| \leq |x|$ such that

$$St_M(\otimes(y,\alpha),|x|) = St_M(\otimes(x,\alpha),|x|).$$

Therefore, after $|x|$ many steps the run of M on $\otimes(x,\alpha)$ coincides with the run on $\otimes(y,\alpha)$. Hence M accepts $\otimes(x,\alpha)$, and x is in L_α.

We define a new indexing $\{H_{\alpha,\beta,\gamma}\}_{(\alpha,\beta,\gamma)\in J}$ for the class $\mathcal{L}$ as follows

$$H_{\alpha,\beta,\gamma} = L_\alpha.$$

Clearly, this indexing is automatic since

$$x \in H_{\alpha,\beta,\gamma} \iff x \in L_\alpha \text{ and } (\alpha,\beta,\gamma) \in J.$$

Now we describe a learner $\mathcal{M}$ that can **Ex**-learn the class $\mathcal{L}$ in the new indexing. Let A be an automaton that recognizes the set J, and let Z be an automaton that rejects all ω-strings. The learner $\mathcal{M}$ will output only automata A and Z in a sequence Z–A–Z (or a subsequence of this). In other words, $\mathcal{M}$ can start outputting automaton Z, then change its mind to A and then again change its mind to Z, after which it will be outputting Z forever.

When an index (α,β,γ) is given to the learner $\mathcal{M}$, it always assumes that β and γ are correctly defined from α. Otherwise, it does not matter which automaton $\mathcal{M}$ will output in the limit, since both A and Z will reject the index (α,β,γ).

At every step s, $\mathcal{M}$ reads the first s inputs $x_1,\dots,x_s$ from the input text. Then $\mathcal{M}$ outputs A if the following conditions hold:

- There exists $n \leq s$ such that $0^n1 \subseteq \beta$.
- For every i with $x_i \neq \#$, x_i belongs to L_α *according to* γ, i.e., $St_M(\otimes(x_i,\alpha),|x_i|) \in \gamma(|x_i|)$.
- For every x with $|x| < n$, if x belongs to L_α according to γ, then $x \in \{x_1,\dots,x_s\}$.

Otherwise, $\mathcal{M}$ outputs Z. This concludes the step s.

Note that $\mathcal{M}$ makes a change from Z to A or from A to Z at most once. Thus it always converges to one of these automata. If the index (α,β,γ) is not in J, then $\mathcal{M}$ always rejects it. If $(\alpha,\beta,\gamma) \in J$, then for every x, we have that $x \in L_\alpha$ according to γ if and only if x is indeed in L_α. Moreover, the set

$$D_n = \{x : |x| < n \text{ and } x \in L_\alpha \text{ according to } \gamma\}$$

is a tell-tale set for L_α, where n is such that $\beta = 0^n1^\omega$.

Let T be the input text. If $\text{content}(T) = H_{\alpha,\beta,\gamma}$, then there is a step $s \geq n$ such that D_n is contained in $\{x_1,\dots,x_s\}$. Therefore, $\mathcal{M}$ will output only A from step s onward. If $\text{content}(T) \neq H_{\alpha,\beta,\gamma}$, then $D_n \not\subseteq \text{content}(T)$ or $\text{content}(T) \not\subseteq H_{\alpha,\beta,\gamma}$. In the first case, $\mathcal{M}$ will output Z on every step. In the second case, there is a step s and an $x_i \in \{x_1,\dots,x_s\}$ such that $x_i \neq \#$ and x_i is not in L_α according to γ. Therefore, $\mathcal{M}$ will output Z from step s onward. This proves the correctness of the algorithm. $\square$

In the previous theorem we showed that a class $\mathcal{L}$ can be **Ex**-learnt in a suitable indexing with the *Reject–Accept–Reject* sequence of mind changes (or a subsequence thereof) if and only if it satisfies Angluin's tell-tale condition. In the rest of this section we will characterize the classes that are **Ex**-learnable with the sequences of mind changes as *Accept–Reject*, *Reject–Accept* and *Accept–Reject–Accept* (or a subsequence thereof). For the ease of notation, we will drop the phrase "or a subsequence thereof" in the following.

Theorem 4.2. *For every automatic class $\mathcal{L}$, the following are equivalent:*

1) *$\mathcal{L}$ can be* **Ex**-*learnt with the Accept–Reject sequence of mind changes in a suitable indexing.*
2) *$\mathcal{L}$ is an inclusion free class, that is, $\forall\, L, L' \in \mathcal{L}\,(L'$ is not a proper subset of $L)$.*

Proof. Suppose that there is a learner $\mathcal{M}$ that **Ex**-learns $\mathcal{L}$ with the sequence of mind changes as *Accept–Reject* in the indexing $\{L_\alpha\}_{\alpha\in I}$, and suppose that there are different sets L_α and L_β such that $L_\alpha \subset L_\beta$. Run the learner $\mathcal{M}$ on the index β and some text for L_α. Since $L_\alpha \neq L_\beta$, there is a step s at which $\mathcal{M}$ changes its mind to *Reject*. Let τ_s be the finite segment of the text seen by $\mathcal{M}$ at step s. Since $L_\alpha \subset L_\beta$, we can extend τ_s to a text T for L_β. Then $\mathcal{M}$ will reject β on text T, which is impossible. Therefore, $\mathcal{L}$ is inclusion free.

Now let $\mathcal{L} = \{L_\alpha\}_{\alpha\in I}$ be an inclusion free class, and let M be a deterministic automaton recognizing $\{(x,\alpha) : x \in L_\alpha\}$. Consider a new set of indices J defined as

$$\begin{aligned}(\alpha,\gamma)\in J \quad \iff \quad & \alpha\in I \text{ and for every } k,\ \gamma(k) = \\ & \{q\in Q_M :\ \exists x\in L_\alpha\,(|x|\le k \text{ and } St_M(\otimes(x,\alpha),k)=q)\}.\end{aligned}$$

Define a new automatic indexing $\{H_{\alpha,\gamma}\}_{(\alpha,\gamma)\in J}$ for the class $\mathcal{L}$ as

$$H_{\alpha,\gamma} = L_\alpha.$$

Let A be an automaton that recognizes the set J, and let Z be an automaton that rejects all ω-strings. The learner $\mathcal{M}$ that **Ex**-learns $\mathcal{L}$ in this new indexing works as follows. At every step s, $\mathcal{M}$ reads the first s inputs $x_1,\ldots,x_s$ from the input text. If every x_i which is not equal to the pause symbol # belongs to $H_{\alpha,\gamma}$ according to γ, i.e., if $St_M(\otimes(x_i,\alpha),|x_i|) \in \gamma(|x_i|)$, then $\mathcal{M}$ outputs A. Otherwise, $\mathcal{M}$ outputs Z.

One can verify that $\mathcal{M}$ **Ex**-learns $\{H_{\alpha,\gamma}\}_{(\alpha,\gamma)\in J}$ with the *Accept–Reject* sequence of mind changes. □

Theorem 4.3. *For every automatic class $\mathcal{L}$, the following are equivalent:*

1) *$\mathcal{L}$ can be* **Ex**-*learnt with the Reject–Accept sequence of mind changes in a suitable indexing.*
2) *For every $L \in \mathcal{L}$ there is a finite $D_L \subseteq L$ such that for every $L' \in \mathcal{L}$, if $D_L \subseteq L'$ then $L' = L$.*

Proof. Suppose that there is a learner $\mathcal{M}$ that **Ex**-learns $\mathcal{L}$ with the sequence of mind changes as *Reject–Accept* in the indexing $\{L_\alpha\}_{\alpha\in I}$. Run $\mathcal{M}$ on an index α and any text for L_α. There must be a step s at which $\mathcal{M}$ changes its mind to *Accept*. Let τ_s be the finite segment of the input text seen by $\mathcal{M}$ at step s, and let $D_\alpha = \mathrm{content}(\tau_s)$. Suppose that there is $L_\beta \neq L_\alpha$ such that $D_\alpha \subseteq L_\beta$. Consider a text T for L_β that extends τ_s. If we run $\mathcal{M}$ on index α and text T, then at step s the learner will change its mind to *Accept*, and after that it will be accepting α forever. On the other hand, $\mathcal{M}$ must eventually reject α since $L_\alpha \neq \text{content}(T)$. Therefore, $\mathcal{L}$ satisfies the condition 2) of the theorem.

Suppose that the class $\mathcal{L} = \{L_\alpha\}_{\alpha \in I}$ satisfies condition 2) of the theorem. Let M be a deterministic automaton that recognizes $\{(x, \alpha) : x \in L_\alpha\}$. The set J of new indices is defined as follows:

$$\begin{aligned}(\alpha, \beta, \gamma) \in J \quad \iff \quad & \alpha \in I,\ \beta = 0^n 1^\omega \text{ for the minimal } n \text{ such that for every } \alpha' \in I \\ & (\{x \in L_\alpha : |x| < n\} \subseteq L_{\alpha'} \ \Rightarrow\ L_{\alpha'} = L_\alpha), \text{ and for every } k \\ & \gamma(k) = \{q \in Q_M :\ \exists x \in L_\alpha\, (|x| \leq k \text{ and } St_M(\otimes(x, \alpha), k) = q)\}.\end{aligned}$$

Using a similar argument as in the proof of Theorem 4.1, one can show that J is automatic. Define a new automatic indexing $\{H_{\alpha,\beta,\gamma}\}_{(\alpha,\beta,\gamma) \in J}$ for the class $\mathcal{L}$ as follows

$$H_{\alpha,\beta,\gamma} = L_\alpha.$$

Let A be an automaton that recognizes the set J, and let Z be an automaton that rejects all ω-strings. The learner $\mathcal{M}$ that **Ex**-learns $\mathcal{L}$ in this new indexing works as follows. At every step s, $\mathcal{M}$ reads the first s inputs $x_1, \ldots, x_s$ from the input text. Then $\mathcal{M}$ outputs A if the following conditions hold:

- There exists $n \leq s$ such that $0^n 1 \subseteq \beta$.
- If $0^n 1 \subseteq \beta$, then for every x with $|x| < n$, if x belongs to L_α according to γ, i.e., $St_M(\otimes(x, \alpha), |x|) \in \gamma(|x|)$, then $x \in \{x_1, \ldots, x_s\}$.

Otherwise, $\mathcal{M}$ outputs Z.

From this description of $\mathcal{M}$ one can see that it **Ex**-learns $\{H_{\alpha,\beta,\gamma}\}_{(\alpha,\beta,\gamma) \in J}$ with the *Reject–Accept* sequence of mind changes. □

Theorem 4.4. *For every automatic class $\mathcal{L}$, the following are equivalent:*

1) *$\mathcal{L}$ can be* **Ex***-learnt with the Accept–Reject–Accept sequence of mind changes in a suitable indexing.*
2) *$\mathcal{L} = \mathcal{H} \cup \mathcal{K}$, where for every $L \in \mathcal{H}$ and $L' \in \mathcal{L}$ $(L' \subseteq L \Rightarrow L' = L)$, and for every $L \in \mathcal{K}$ there is a finite $D_L \subseteq L$ such that for every $L' \in \mathcal{L}$ $(D_L \subseteq L' \Rightarrow L' = L)$.*

Proof. Suppose that there is a learner $\mathcal{M}$ that **Ex**-learns $\mathcal{L}$ with the *Accept–Reject–Accept* sequence of mind changes in the indexing $\{L_\alpha\}_{\alpha \in I}$. Define $\mathcal{H}$ and $\mathcal{K}$ as follows:

$$\mathcal{H} = \{L_\alpha :\ \text{every automaton output by } \mathcal{M} \text{ on index } \alpha \text{ and any text for } L_\alpha \text{ accepts } \alpha\}$$

and

$$\begin{aligned}\mathcal{K} = \{L_\alpha : \ & \text{there is a text } T \text{ for } L_\alpha \text{ such that the learner } \mathcal{M} \text{ has a } \textit{Reject–Accept} \text{ or} \\ & \textit{Accept–Reject–Accept} \text{ pattern of mind changes when it processes } \alpha \text{ and } T\}.\end{aligned}$$

Suppose that there are different $L_\alpha \in \mathcal{L}$ and $L_\beta \in \mathcal{H}$ such that $L_\alpha \subset L_\beta$. Run the learner $\mathcal{M}$ on index β and some text for L_α. There must be a step s at which $\mathcal{M}$ outputs an automaton rejecting β. Let τ_s be the finite segment of the text seen by $\mathcal{M}$ at step s. Since $L_\alpha \subset L_\beta$, we can extend τ_s to a text T for L_β. Now $\mathcal{M}$ outputs an automaton rejecting β when it processes β and T. This contradicts our definition of $\mathcal{H}$.

Suppose that $L_\alpha \in \mathcal{K}$ and let T be a text for L_α such that the learner $\mathcal{M}$ has a pattern of mind changes *Reject–Accept* or *Accept–Reject–Accept* when it processes α and T. Run $\mathcal{M}$ on the index α and the text T. Let s be the step at which $\mathcal{M}$ changes its mind from *Reject* to *Accept*, and let τ_s be the finite segment of text T seen by this step. Define $D_\alpha = \text{content}(\tau_s)$.

Suppose that there is $L_\beta \in \mathcal{L}$ such that $L_\beta \neq L_\alpha$ and $D_\alpha \subseteq L_\beta$. Consider a text T for L_β that extends τ_s. If we run $\mathcal{M}$ on index α and text T, then at step s the learner will change its mind from *Reject* to *Accept*, and after that it will be accepting α forever. On the other hand, $\mathcal{M}$ must eventually reject α since $L_\alpha \neq \text{content}(T)$. Therefore, $\mathcal{L}$ satisfies the condition 2) of the theorem.

Now suppose that the class $\mathcal{L} = \{L_\alpha\}_{\alpha \in I}$ satisfies condition 2) of the theorem. Let M be a deterministic automaton that recognizes $\{(x, \alpha) : x \in L_\alpha\}$. The set J of new indices is defined as follows:

$$
\begin{aligned}
(\alpha, \beta, \gamma) \in J \quad \iff \quad & \alpha \in I,\ \beta = 0^n 1^\omega \text{ for the minimal } n \text{ such that for every } \alpha' \in I \\
& (\{x \in L_\alpha : |x| < n\} \subseteq L_{\alpha'} \ \Rightarrow\ L_{\alpha'} = L_\alpha), \text{ and for every } k \\
& \gamma(k) = \{q \in Q_M :\ \exists x \in L_\alpha\, (|x| \leq k \text{ and } St_M(\otimes(x, \alpha), k) = q)\}.
\end{aligned}
$$

Again, the set J is automatic, and we can define a new automatic indexing $\{H_{\alpha,\beta,\gamma}\}_{(\alpha,\beta,\gamma)\in J}$ for the class $\mathcal{L}$ as follows

$$H_{\alpha,\beta,\gamma} = L_\alpha.$$

Let A be an automaton that recognizes the set J, and let Z be an automaton that rejects all ω-strings. The learner $\mathcal{M}$ that **Ex**-learns $\mathcal{L}$ in this new indexing works as follows. At every step s, $\mathcal{M}$ reads the first s inputs $x_1, \ldots, x_s$ from the input text. Then $\mathcal{M}$ outputs A or Z according to the following rules:

Case A: There is no $n \leq s$ such that $0^n 1 \subseteq \beta$. In this case $\mathcal{M}$ outputs A if every x_i which is different from $\#$ belongs to L_α according to γ. Otherwise, $\mathcal{M}$ outputs Z.

Case B: There exists $n \leq s$ such that $0^n 1 \subseteq \beta$. In this case $\mathcal{M}$ outputs A if

$$\forall x\, ((|x| < n \text{ and } x \in L_\alpha \text{ according to } \gamma)\ \rightarrow\ x \in \{x_1, \ldots, x_s\}).$$

Otherwise, $\mathcal{M}$ outputs Z.

It is clear that $\mathcal{M}$ has an *Accept–Reject–Accept* sequence of mind changes (or a subsequence thereof) for any index $(\alpha, \beta, \gamma) \in J$ and any text T with $\text{content}(T) \in \mathcal{L}$. If $\mathcal{M}$ always stays in Case A, then $H_{\alpha,\beta,\gamma} = L_\alpha$ is not in $\mathcal{K}$ and hence $L_\alpha \in \mathcal{H}$. By construction, $\mathcal{M}$ eventually accepts (α, β, γ) if and only if $\text{content}(T) \subseteq L_\alpha$. But since $L_\alpha \in \mathcal{H}$, we have that $\text{content}(T) \subseteq L_\alpha$ implies $\text{content}(T) = L_\alpha$.

If at some step the learner $\mathcal{M}$ is in Case B, then $H_{\alpha,\beta,\gamma} = L_\alpha \in \mathcal{K}$. By construction, $\mathcal{M}$ eventually accepts (α, β, γ) if and only if $D_\alpha \subseteq \text{content}(T)$, where $D_\alpha = \{x \in L_\alpha : |x| < n\}$. By the definition of β, $D_\alpha \subseteq \text{content}(T)$ implies $L_\alpha = \text{content}(T)$. □

5 Blind Learning

Blind learning is distinguished from models of learning described in the previous sections in that a learner itself does not see the index tape. So the learner has to encode all the necessary information into a sequence of automata which determines in the limit whether an index is correct or incorrect. In the case of behaviourally correct learning, this can be done by coding more and more finite information into such a sequence in a way that every incorrect index is eventually rejected (but the point from which on this happens depends on an index). In the case of explanatory learning, this turns out to be impossible. However if a class is countable, then we can simulate a traditional learner (for a countable class) and encode its output conjecture into an ω-automaton which then checks whether the index provided is equivalent to the current output of the traditional learner. In some sense, this is the best that one can do, as all blind explanatorily learnable classes are countable.

Theorem 5.1. *If a class $\mathcal{L} = \{L_\alpha\}_{\alpha \in I}$ satisfies Angluin's tell-tale condition, then $\mathcal{L}$ is* **BlindBC***-learnable.*

Proof. We describe an algorithm for a **BlindBC**-learner $\mathcal{M}$.

At step s, the learner reads the first s inputs $x_1, \ldots, x_s$ from the input text. If every x_i is equal to the pause symbol #, then the learner outputs an automaton which accepts exactly the indices of $\emptyset$. Otherwise, let $z_1^s, \ldots, z_t^s$ be such that $z_1^s <_{llex} \cdots <_{llex} z_t^s$ and $\{z_1^s, z_2^s, \ldots, z_t^s\} = \{x_1, x_2, \ldots, x_s\} - \{\#\}$. For every k with $1 \le k \le t$, let A_k^s be an automaton such that

$$\begin{aligned} A_k^s \text{ accepts } \alpha \quad \iff \quad & \alpha \in I,\ (\{x_1, \ldots, x_s\} - \{\#\}) \subseteq L_\alpha, \\ & \{x_1, \ldots, x_s\} \cap \{x : x \le_{llex} z_k^s\} = L_\alpha \cap \{x : x \le_{llex} z_k^s\}, \\ & \text{and } L_\alpha \cap \{x : x \le_{llex} z_k^s\} \text{ is a tell-tale set for } L_\alpha. \end{aligned}$$

Such an A_k^s exists since the property of being a tell-tale set is first-order definable from other automatic relations as described in the beginning of the proof of Theorem 3.1. Finally, in the end of step s, $\mathcal{M}$ outputs an automaton A^s such that

$$L(A^s) = \bigcup_{1 \le k \le t} L(A_k^s).$$

To verify that the algorithm is correct, we need to show that for every input text T with $\text{content}(T) \in \mathcal{L}$ and for every index α

a) if $\alpha \in I$ and $L_\alpha = \text{content}(T)$, then A^s accepts α for almost all s,
b) if $\alpha \in I$ and $L_\alpha \ne \text{content}(T)$ or if $\alpha \notin I$, then A^s rejects α for almost all s.

First, suppose that $L_\alpha = \text{content}(T)$. Since $\mathcal{L}$ satisfies Angluin's tell-tale condition, there are s_0 and k such that for all $s \ge s_0$

$$L_\alpha \cap \{x : x \le_{llex} z_k^s\} \text{ is a tell-tale set for } L_\alpha.$$

Let $s_1 \ge s_0$ be such that for every $s \ge s_1$

$$\{x_1, \ldots, x_s\} \cap \{x : x \le_{llex} z_k^s\} = L_\alpha \cap \{x : x \le_{llex} z_k^s\}.$$

Then, by definition, A_k^s accepts α for all $s \ge s_1$. Therefore, A^s accepts α for all $s \ge s_1$.

If $\alpha \notin I$, then every A^s rejects α (note that our definitions of learning do not place any requirements on the learner when $\alpha \notin I$; this point is just for emphasis). So, suppose that $\alpha \in I$ and $L_\alpha \neq \text{content}(T)$. If $\exists x \in \text{content}(T) \setminus L_\alpha$, then for some s_0 we have that $x \in \{x_1, \ldots, x_s\}$ for all $s \geq s_0$. Therefore, for all $s \geq s_0$, $(\{x_1, \ldots, x_s\} - \{\#\}) \not\subseteq L_\alpha$ and A^s rejects α. Suppose that content(T) is a proper subset of L_α. Note that for every s and k, if $L_\alpha \cap \{x : x \leq_{llex} z_k^s\}$ is a tell-tale set for L_α, then

$$\{x_1, \ldots, x_s\} \cap \{x : x \leq_{llex} z_k^s\} \neq L_\alpha \cap \{x : x \leq_{llex} z_k^s\}.$$

Otherwise, content(T) would be a proper subset of L_α containing a tell-tale set for L_α, which is impossible. So, every A_k^s and hence every A^s rejects α. □

Theorem 5.2. *For every class $\mathcal{L} = \{L_\alpha\}_{\alpha \in I}$, the following are equivalent*

1) *$\mathcal{L}$ is* **BlindEx***-learnable.*
2) *$\mathcal{L}$ is* **BlindFEx***-learnable.*
3) *$\mathcal{L}$ is at most countable and satisfies Angluin's tell-tale condition.*

Proof. It is obvious that **BlindEx**-learnable class is **BlindFEx**-learnable. Suppose that $\mathcal{L} \in$ **BlindFEx**; then Fact 2.13 implies that $\mathcal{L}$ satisfies Angluin's tell-tale condition. We will show that $\mathcal{L}$ is countable.

Let $\mathcal{M}$ be a **BlindFEx**-learner for $\mathcal{L}$. Thus for every $L \in \mathcal{L}$ and every input text T with content$(T) = L$, the learner $\mathcal{M}$ outputs at least one automaton A_L infinitely often. Since $\mathcal{M}$ is blind, A_L must accept all indices α with $L_\alpha = L$ and reject all indices β with $L_\beta \neq L$. If L and L' are two different sets from $\mathcal{L}$, then $A_L \neq A_{L'}$. Since there are only countably many different automata, the class $\mathcal{L}$ is at most countable.

Suppose that $\mathcal{L}$ is countable and satisfies Angluin's tell-tale condition. Consider the following equivalence relation on the set I of indices for $\mathcal{L}$:

$$\alpha \sim \beta \quad \text{if and only if} \quad \forall x\,(x \in L_\alpha \leftrightarrow x \in L_\beta).$$

This equivalence relation is automatic since it is first-order definable from automatic relations. By assumption, it has countable index. As Bárány, Kaiser and Rubin [2] showed, every automatic equivalence relation of countable index has a countable automatic set of representatives. Let $J \subseteq I$ be a set of such representatives.

It is well-known that every automatic set of ω-strings is a finite union of sets of the form $V \cdot U^\omega$, where V and U are automatic sets of finite strings (e.g., see [11]). If the set is countable, then U contains only a single string u. Therefore, we have that $J = \bigcup_{i=1}^k V_i \cdot \{u_i\}^\omega$ for some automatic sets V_i and finite strings u_i.

We now define an automatic indexing of the class $\mathcal{L}$ by finite strings. Let Σ be the alphabet of the set I and let Γ be the alphabet of the sets L_α. Consider an expanded alphabet $\Sigma' = \Sigma \cup \{1, \ldots, k\}$ (we assume here that Σ does not contain $\{1, \ldots, k\}$). A set G of new indices will be

$$G = \{\, vi : \ v \in V_i \text{ and } i \in \{1, \ldots, k\}\,\}.$$

Note that G is automatic. The new indexing $\{H_w\}_{w \in G}$ of $\mathcal{L}$ is defined as follows: for every $vi \in G$, let

$$H_{vi} = L_{vu_i^\omega}.$$

We need to show that the relation $R = \{(x, w) : x \in H_w\}$ is automatic. Let M be a deterministic Muller automaton that recognizes the relation $\{(x, \alpha) : x \in L_\alpha\}$. A finite automaton A that recognizes R can be defined informally as follows. On input $\otimes(x, vi)$, A simulates M on the input $\otimes(x, vu_i^\omega)$. After processing its input, A accepts it if and only if there is an accepting run of M on $\otimes(x, vu_i^\omega)$ (that is, A replaces i by u_i^ω in its simulation). Thus,

$$A \text{ accepts } \otimes(x, vi) \iff M \text{ accepts } \otimes(x, vu_i^\omega).$$

Below is a formal definition of A.

Suppose that $M = (Q^M, (\Gamma \cup \Sigma \cup \{\#\})^2, q_0^M, T^M)$. For $i = 1, \dots, k$, let $u_i = u_{i,1} \dots u_{i,n_i}$, where n_i is the length of u_i. Then $A = (Q, (\Gamma \cup \Sigma' \cup \{\#\})^2, q_0, T, F)$, where

1) $Q = \{\, (q, i, j) : \; q \in Q^M, \; 0 \le i \le k$, if $i = 0$ then $j = 0$, and if $i > 0$ then $1 \le j \le n_i \,\}$.
2) $q_0 = (q_0^M, 0, 0)$.
3) The transition function T is defined as follows:
   a) for every $a \in \Gamma \cup \{\#\}$ and $b \in \Sigma$,
   $$T((q, 0, 0),\, (a, b)) = (T^M(q,\, (a, b)), 0, 0);$$
   b) for every $a \in \Gamma \cup \{\#\}$ and $i \in \{1, \dots, k\}$,
   $$T((q, 0, 0),\, (a, i)) = (T^M(q,\, (a, u_{i,1})), i, 1);$$
   c) for every $a \in \Gamma$, $i \in \{1, \dots, k\}$ and $j \in \{1, \dots, n_i\}$,
   $$T((q, i, j),\, (a, \#)) = (T^M(q,\, (a, u_{i,j+1})), i, j + 1),$$
   where it is assumed that $n_i + 1 = 1$.
4) The final states are defined as

$$F = \{\, (q, i, j) \in Q : \; i > 0, \text{ and there exists an accepting run of } M \text{ on the string } \otimes\big(\#^\omega, (u_i^{[j]})^\omega\big) \text{ starting from } q \,\},$$

where $u_i^{[j]}$ is the cyclic shift of u_i by j symbols, i.e.,

$$u_i^{[j]} = u_{i,j+1} \dots u_{i,n_i} u_{i,1} \dots u_{i,j}.$$

Note that the final states F of the automaton A can be computed effectively.

Since $\{H_w\}_{w \in G}$ is automatic and satisfies Angluin's tell-tale condition, there is a recursive learner $\mathcal{M}'$ such that, on any input text T for H_w (where $w \in G$), $\mathcal{M}'$ converges on T to an index w' such that $H_{w'} = H_w$ (see [10]; this is traditional **Ex**-learning of countable automatic families satisfying Angluin's tell-tale condition). For every $w \in G$, let A_w be an automaton such that

$$L(A_w) = \{\alpha \in I : \; \forall x\, (x \in L_\alpha \leftrightarrow x \in H_w)\}.$$

Such an A_w exists since $L(A_w)$ is first-order definable from automatic relations. Now the **BlindEx**-learner $\mathcal{M}$ for the class $\mathcal{L} = \{L_\alpha\}_{\alpha \in I}$ acts as follows: on an input text T for some $L \in \mathcal{L}$, it simulates the work of $\mathcal{M}'$, and whenever $\mathcal{M}'$ outputs an index $w \in G$, the learner $\mathcal{M}$ outputs the automaton A_w.

Since $\mathcal{M}'$ converges to an index w such that $H_w = \text{content}(T)$, we have that $\mathcal{M}$ converges to the automaton A_w such that $L(A_w) = \{\alpha \in I : L_\alpha = \text{content}(T)\}$. Therefore, the class $\mathcal{L}$ is **BlindEx**-learnable. □

The following corollary summarizes the main results from this and the previous sections.

Corollary 5.3. *For every automatic class $\mathcal{L}$, the following are equivalent:*

1) *$\mathcal{L}$ satisfies Angluin's tell-tale condition.*
2) *$\mathcal{L}$ is* **BC***-learnable.*
3) *$\mathcal{L}$ is* **BlindBC***-learnable.*
4) *$\mathcal{L}$ is* **FEx***-learnable.*
5) *$\mathcal{L}$ is* **Ex***-learnable in a suitable indexing.*

Proof. The implications 3) $\Rightarrow$ 2) and 4) $\Rightarrow$ 2) are trivial; 2) $\Rightarrow$ 1) and 5) $\Rightarrow$ 1) follow from Fact 2.13; 1) $\Rightarrow$ 3) follows from Theorem 5.1; 1) $\Rightarrow$ 4) follows from Theorem 3.1; and 1) $\Rightarrow$ 5) follows from Theorem 4.1. □

6 Partial Identification

Partial identification is, in the traditional setting of inductive inference, a learning criterion where the learner outputs on every text of an r.e. language infinitely many (not necessarily distinct) hypotheses such that exactly one hypothesis occurs infinitely often and that hypothesis is correct. There is a recursive learner succeeding on all r.e. sets, hence this concept is omniscient in the traditional setting [13]. Also in our model, every automatic class is partially identifiable.

Theorem 6.1. *Every automatic class with any given automatic indexing is* **Part***-learnable.*

Proof. Consider an automatic indexing $\{L_\alpha\}_{\alpha \in I}$ for a class $\mathcal{L}$. Let M be an automaton recognizing the relation '$x \in L_\alpha$', and let $\equiv_{M,\alpha}$ and $\equiv_{M,\alpha,s}$ be the relations defined in the proof of Theorem 3.1. For every pair of strings (x, y) with $x <_{llex} y$, let $Z_{(x,y)}$ be an automaton that rejects all inputs. For every $k \geq 1$ and every tuple $(x_1, \ldots, x_k)$ with $x_1 <_{llex} \cdots <_{llex} x_k$, let $A_{(x_1,\ldots,x_k)}$ be an automaton that accepts an ω-string α if and only if

$$\forall y \, (y \in L_\alpha \iff y \equiv_{M,\alpha} x_i \text{ for some } i \in \{1, \ldots, k\}).$$

We assume that all the automata defined above are different from each other. Extend the ordering $\leq_{llex}$ to pairs of strings as follows: $(x', y') \leq_{llex} (x, y)$ if and only if one of the following conditions is satisfied:

1) $\max\{|x'|, |y'|\} < \max\{|x|, |y|\}$,
2) $\max\{|x'|, |y'|\} = \max\{|x|, |y|\}$ and $x' <_{llex} x$,
3) $\max\{|x'|, |y'|\} = \max\{|x|, |y|\}$, $x = x'$ and $y' \leq_{llex} y$.

Let $\mathcal{M}$ be a learner constructed to satisfy the following properties:

1) $\mathcal{M}$ outputs the automaton $Z_{(x,y)}$ on index α and text T at least n times if and only if there exists $s \geq n$ such that

– $x <_{llex} y$ and $x \equiv_{M,\alpha,s} y$,
– $|\{x, y\} \cap \text{content}(\tau_s)| = 1$, where τ_s is the initial segment of T of length s,
– there is no $(x', y') <_{llex} (x, y)$ for which the above two properties hold.

2) $\mathcal{M}$ outputs the automaton $A_{(x_1,\ldots,x_k)}$ on index α and text T at least n times if and only if there exists $s \geq n$ such that
– for every $i \in \{1, \ldots, k\}$ and every $y <_{llex} x_i$ we have that $y \not\equiv_{M,\alpha,s} x_i$,
– $\{x_1, \ldots, x_k\} \subseteq \text{content}(\tau_s)$,
– for every $z \notin \{x_1, \ldots, x_k\}$ with $|z| \leq n$, if $\forall y <_{llex} z \ (y \not\equiv_{M,\alpha,s} z)$ then $z \notin \text{content}(\tau_s)$,
– for every x, y such that $\max\{|x|, |y|\} \leq n$, if $x <_{llex} y$ and $x \equiv_{M,\alpha,s} y$ then $|\{x, y\} \cap \text{content}(\tau_s)| \neq 1$.

It is not hard to verify that if the learner $\mathcal{M}$ satisfies the above properties, then for any α and T:

a) $\mathcal{M}$ outputs $Z_{(x,y)}$ infinitely often on α, T if and only if (x, y) is the $\leq_{llex}$ least pair such that $x <_{llex} y$, $x \equiv_{M,\alpha} y$ and $|\{x, y\} \cap \text{content}(T)| = 1$.
b) $\mathcal{M}$ outputs $A_{(x_1,\ldots,x_k)}$ infinitely often on α, T if and only if $x_1, \ldots, x_k$ are exactly those $\leq_{llex}$ least representatives of equivalence classes of $\equiv_{M,\alpha}$ which belong to content(T), and there is no (x, y) such that $x <_{llex} y$, $x \equiv_{M,\alpha} y$ and $|\{x, y\} \cap \text{content}(T)| = 1$.

Now, if content(T) is not equal to the union of equivalence classes of $\equiv_{M,\alpha}$, then $\mathcal{M}$ outputs only $Z_{(x,y)}$ infinitely often for some (x, y), and it rejects the index α. Otherwise, $\mathcal{M}$ outputs only $A_{(x_1,\ldots,x_k)}$ infinitely often, where $x_1, \ldots, x_k$ are the $\leq_{llex}$ least representatives of the equivalence classes belonging to content(T). By definition, $A_{(x_1,\ldots,x_k)}$ accepts index α if and only if L_α is the union of the equivalence classes of $x_1, \ldots, x_k$. The latter is equivalent to $L_\alpha = \text{content}(T)$ by the property b) above. □

Theorem 6.2. *A class $\mathcal{L} = \{L_\alpha\}_{\alpha \in I}$ is in* **BlindPart** *if and only if it is at most countable.*

Proof. First, we show that if $\mathcal{L} \in$ **BlindPart**, then it is at most countable. Let $\mathcal{M}$ be a **BlindPart**-learner for $\mathcal{L}$. Fix a set $L \in \mathcal{L}$ and some text T for L. The learner $\mathcal{M}$ outputs exactly one automata infinitely often when processing the text T. Let A be such an automaton. Since $\mathcal{M}$ is blind, A must accept only those α for which $L_\alpha = L$. Since there are only countably many different automata, the class $\mathcal{L}$ is at most countable.

To prove the other implication, assume that $\mathcal{L}$ is at most countable. In this case we can construct a new automatic indexing $\{H_w\}_{w \in G}$ for $\mathcal{L}$ by finite strings as shown in the proof of Theorem 5.2. Moreover, we can choose this indexing to be one-to-one. For every $w \in G$, let A_w be an automaton that recognizes the set $\{\alpha \in I : L_\alpha = H_w\}$.

The **BlindPart**-learner $\mathcal{M}$ works as follows. At every step s, $\mathcal{M}$ reads the first s inputs $x_1, \ldots, x_s$ from the input text T, and for every $w \in G$ with $|w| \leq s$, it computes the coincidence between $\{x_1, \ldots, x_s\}$ and H_w at step s, that is,

$$C(w, s) = \max\{n : \ n \leq s \text{ and for every string } x \text{ with } |x| \leq n \\ (x \in \{x_1, \ldots, x_s\} \iff x \in H_w)\}.$$

If there exists a $w \in G$ with $|w| \leq s$ and $C(w, s) > C(w, s-1)$, then $\mathcal{M}$ outputs A_w for the $\leq_{llex}$ least such w. Otherwise, $\mathcal{M}$ does not produce an output at step s.

To verify that the algorithm is correct, let T be a text for a set $L \in \mathcal{L}$ and let w_0 be an index such that $H_{w_0} = L$. Since the indexing $\{H_w\}_{w \in G}$ is one-to-one, we have that $\lim_s C(w_0, s) = \infty$, but for every $w' \neq w_0$, $\lim_s C(w', s) < \infty$. Thus, every $A_{w'}$ with $w' \neq w_0$ will be output only finitely often. Let s_0 be a step by which all $C(w', s)$ with $w' <_{llex} w_0$ have reached their limit. Then at every step $s \geq s_0$ such that $C(w_0, s) > C(w_0, s-1)$, $\mathcal{M}$ outputs A_{w_0}. Therefore, A_{w_0} is output infinitely often and by definition $L(A_{w_0}) = \{\alpha \in I : L_\alpha = H_{w_0} = L\}$. $\square$

References

[1] Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45(2):117–135, 1980.

[2] Vince Bárány, Łukas Kaiser and Sasha Rubin. Cardinality and counting quantifiers on omega-automatic structures. In *Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science, STACS 2008*, pages 385–396, 2008.

[3] Janis Bārzdiņš. Two theorems on the limiting synthesis of functions. *Theory of Algorithms and Programs*, 1:82–88, Latvian State University, Riga, Latvia, 1974.

[4] Achim Blumensath and Erich Grädel. Automatic structures. In *15th Annual IEEE Symposium on Logic in Computer Science (Santa Barbara, CA, 2000)*, pages 51–62. IEEE Computer Society Press, Los Alamitos, CA, 2000.

[5] Achim Blumensath and Erich Grädel. Finite presentations of infinite structures: automata and interpretations. *Theory of Computing Systems*, 37(6):641–674, 2004.

[6] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[7] J. Richard Büchi. On a decision method in restricted second order arithmetic. In *Logic, Methodology and Philosophy of Science (Proceedings 1960 International Congress)*, pages 1–11. Stanford University Press, Stanford, California, 1962.

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

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

[10] Sanjay Jain, Qinglong Luo and Frank Stephan. Learnability of automatic classes. Technical Report TRA1/09, School of Computing, National University of Singapore, 2009. Preliminary version to appear in *4th International Conference on Language and Automata Theory and Applications* (LATA 2010).

[11] Bakhadyr Khoussainov and Anil Nerode. *Automata theory and its applications.* Birkhäuser Boston, Inc., Boston, MA, 2001.

[12] Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In *Logic and Computational Complexity (Indianapolis, IN, 1994)*, volume 960 of *Lecture Notes in Computer Science*, pages 367–392. Springer, Berlin, 1995.

[13] Daniel N. Osherson, Micheal Stob and Scott Weinstein. *Systems that learn. An introduction to learning theory for cognitive and computer scientists.* Bradford Book—MIT Press, Cambridge, MA, 1986.

[14] Moshe Y. Vardi. The Büchi complementation saga. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science, STACS 2007*, volume 4393 of *Lecture Notes in Computer Science*, pages 12–22. Springer, Berlin, 2007.