Counting Extensional Differences in BC-Learning

Sanjay Jain [a,1] Frank Stephan [b,2] Sebastiaan A. Terwijn [c,3]

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

[b]*Mathematisches Institut, Universität Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany. Email: fstephan@math.uni-heidelberg.de*

[c] *Institute for Algebra and Computational Mathematics, Technical University of Vienna, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria. Email: terwijn@logic.at*

Abstract

Let BC be the model of behaviourally correct function learning as introduced by Bārzdins [4] and Case and Smith [8]. We introduce a mind change hierarchy for BC, counting the number of extensional differences in the hypotheses of a learner. We compare the resulting models BC_n to models from the literature and discuss confidence, team learning, and finitely defective hypotheses. Among other things, we prove that there is a tradeoff between the number of semantic mind changes and the number of anomalies in the hypotheses. We also discuss consequences for language learning. In particular we show that, in contrast to the case of function learning, the family of classes that are confidently BC-learnable from text is not closed under finite unions.

Keywords. Models of grammar induction, inductive inference, behaviourally correct learning.

1 Introduction

Gold [10] introduced an abstract model of learning computable functions, where a learner receives increasing amounts of data about an unknown func-

tion and outputs a sequence of hypothesis that has to converge to a single explanation, i.e. a program, for the function at hand. This concept of explanatory or Ex-learning has been widely studied [8,10,11,15]. A recurring theme is the question how often the learner can change its hypothesis and how conscious it is of this process: does the learner know when it has converged and how fast does the learner see when new data requires the hypothesis to be changed. Gold [10] already observed that a learner who knows when the correct hypothesis has been found is quite restricted: such a learner can wait until it has the correct hypothesis and then output a single but correct guess. Therefore such a learner can never learn a dense class of functions, which requires one to be able to withdraw and change to a new hypothesis at arbitrary late time points, as in the model Ex.

Another well-studied paradigm is the model BC of behaviourally correct learning [4,8]. The difference with the Ex-model lies in the notion of convergence: Whereas in Ex the syntax of the hypotheses of the learner is required to converge, i.e. convergence is *intensional*, in BC the semantics of the hypotheses should converge, i.e. convergence is *extensional*. Bārzdins [4] showed that behaviourally correct learners can learn classes on which no Ex-learner succeeds. BC-learners are quite powerful: Steel [8] noticed that the concept of syntactic convergence to an almost everywhere correct hypothesis can be covered by an error-free BC-learner. Furthermore, Harrington [8] showed that a further generalization of BC-learners, namely those which almost always output finite variants of the function to be learned, can learn all recursive functions.

There are many models of learning in which the number of changes in hypothesis, also called mind changes, is counted. Previous studies focussed mainly on intermediate notions employing syntactic convergence. In particular Bārzdins and Freivalds [5] initiated the analysis of Ex-learning with a bound on the number of mind changes. Freivalds and Smith [9] generalized this concept by using recursive ordinals which are counted down recursively at every mind change. Just as it is interesting to study syntactic mind changes, we find it interesting to explore semantic mind changes. In Section 3 we introduce the models BC_n (for n being a natural number) where the BC-learner may make at most n semantic mind changes on any function to be learned. It is shown that the classes BC_n form a proper hierarchy that is incomparable to Ex-learning.

Ambainis, Jain and Sharma [1] showed that a class of functions is Ex-learnable with a recursive ordinal number of mind changes if and only if it can be learned by a machine which converges on every function, even on the nonrecursive ones, to some hypothesis. Following Osherson, Stob and Weinstein [17, Section 4.6.2], we call a learner that converges on all functions *confident*. This notion can be generalized to BC: A BC-learner is confident if it converges semantically on every function. Before we define ordinal mind change bounds for BC, we take instead the characterization of ConfEx as an alternative starting point

and study ConfBC. In Section 4, we show among other things that the result that all classes Ex_n are in confident Ex also holds in the case of semantic convergence: Every BC_n-learnable class has a confident BC-learner. At the end we show how to introduce ordinal mind change bounds for BC-learning and note that this concept is equivalent to the notion ConfBC.

In Section 5 we consider hypotheses which are finitely defective. The more noticeable difference with the Ex case is that here there is a tradeoff between anomalies and mind changes. We prove that BC_1, the first nontrivial level of the BC_n hierarchy (since BC_0 coincides with Ex_0), is not contained in OEx^*, a learning criterion from Case and Smith [8]. This improves a result from [8].

Finally, in Section 6 we discuss consequences for grammatical inference. In [10] Gold also introduced a model of learning recursively enumerable sets (in this context also called *languages*), which is more general than the model of learning recursive functions. The negative results obtained in the previous sections for function learning immediately imply their counterparts for language learning. In this section we discuss the positive counterparts. In contrast to the case of function learning we show that the family of classes that are confidently BC-learnable from text is not closed under finite unions. We do this by constructing a certain class of finite thickness that also shows that a result from [22] is optimal.

2   Preliminaries and Notation

We will use the following notation. For a function f, $f[n]$ denotes the string $f(0)f(1)f(2)\ldots f(n-1)$. Our recursion theoretic notation is standard and follows Odifreddi [14] and Soare [20]. φ denotes a standard acceptable programming system. φ_e is the e-th partial recursive function, and $\varphi_{e,s}(x)$ is the result of running φ_e for s steps on input x. $\mathbb{N}$ is the set of natural numbers. $\langle .,.\rangle$ denotes a standard pairing function. For a string σ, $|\sigma|$ is the length of σ.

We recall the following definitions. A recursive function M from finite sequences of natural numbers to $\mathbb{N}$, Ex-*learns* (see [10]) a recursive function f if $k = \lim_{n\to\infty} M(f[n])$ exists and is a code for f, i.e. $\varphi_k = f$. We say that M Ex-learns a class $\mathcal{C}$ of recursive functions if and only if M Ex-learns each function in the class. M BC-*learns* (see [4,8]) a recursive function f, if for almost every n, $M(f[n])$ is a code for f, i.e. $\varphi_{M(f[n])} = f$. We say that M BC-learns a class $\mathcal{C}$ of recursive functions if and only if M BC-learns each function in the class. Ex and BC denote the families of classes that are learnable by a recursive Ex and BC learner, respectively.

3

In the literature on inductive inference, it is customary to allow a learner to output initially the symbol "?", that does not count as a numerical hypothesis. This is relevant when counting the number of mind changes that a learner makes on given input data. We say that a learner M makes a *mind change* on f at $n + 1$, if $M(f[n]) \neq ?$ and $M(f[n]) \neq M(f[n+1])$. A class of recursive functions $\mathcal{C}$ is in Ex_m, if there is a recursive learner that learns every $f \in \mathcal{C}$ by making at most m mind changes on f.

We will also consider team learning [16,19]. Recall that for a learning criterion $\mathcal{I}$, a class $\mathcal{A}$ is in $[m, n]\mathcal{I}$, if there is a *team* consisting of n learners such that, for every $f \in \mathcal{A}$, at least m of these learners $\mathcal{I}$-learn f.

3   Semantic mind changes

It is clear that the notion of mind change as defined above is not useful for the study of the model of BC-learning, since in this model the inductive inference machine does not have to converge to a particular code for the input function but may infinitely often output a different code, as long as in the limit these codes are for the input function. In other words, in the limit the outputs of the function may differ syntactically but *semantically* they must be the same. This brings us to define a notion of mind change for BC-learning as follows.

Definition 1 A machine M BC_n-*learns* a recursive function f (or: M BC-learns f with at most n *semantic mind changes*) if M BC-learns f such that the cardinality of the set $\{m : M(f[m]) \neq ? \wedge \varphi_{M(f[m])} \neq \varphi_{M(f[m+1])}\}$ is at most n.
M BC_n-learns a class $\mathcal{C}$ of recursive functions, if M BC_n-learns each function in $\mathcal{C}$.
BC_n denotes the family of classes that can be BC_n-learned by some recursive learner.

That is, the machine M is allowed only n *semantic* mind changes, i.e. a change of output from e_0 to e_1 such that $\varphi_{e_0} \neq \varphi_{e_1}$. Here, as in the case of Ex_n, an initial sequence of empty hypotheses "?" is allowed. In the following, when we speak about mind changes it will depend on the model under consideration what we mean: If the model is defined using the basic model Ex we will always mean 'mind change' in the previously defined, syntactical, sense and if the model is a variant of BC we will always use the semantic meaning of the word mind change. We now state the basic properties of the model BC_n and show how it relates to the other models.

Theorem 2   (a) $\mathrm{BC}_0 = \mathrm{Ex}_0$.
  (b) $\mathrm{Ex}_n \subset \mathrm{BC}_n$ *for* $n \geq 1$.

(c) *For every* $n \in \mathbb{N}$ *it holds that* $\mathrm{Ex}_{n+1} \not\subseteq \mathrm{BC}_n$.

(d) $\mathrm{Ex} \not\subseteq \bigcup_{n \in \mathbb{N}} \mathrm{BC}_n$.

(e) BC_1 *is not contained in* Ex.

Proof. (a) $\mathrm{Ex}_0 \subseteq \mathrm{BC}_0$ by definition. To Ex_0-learn a class $\mathcal{C}$, which is BC_0-learned by a machine M: on any input function, simply output the first hypothesis of M that is unequal to "?". For functions in $\mathcal{C}$, since M is not permitted to change the hypothesis semantically, the first hypothesis must be correct.

The strictness of the inclusion in (b) follows from (e). Items (c) and (d) can be proven by a well-known argument used in Theorem 14 in order to obtain a more general result. Item (e) will be proven in Theorem 19. □

The following two propositions are useful for us. The proofs, which are easy diagonalizations similar to the ones found in [8], are left to the reader.

Proposition 3 (Based on [8]) *Let* $n \in \mathbb{N}$. *Consider the classes*

$$\mathcal{C}_1^n = \{f : f(0) = n+1 \text{ and } f(x+1) \leq f(x) \text{ for all } x\},$$
$$\mathcal{C}_2^n = \{f : \mathrm{card}(\{x : f(x) \neq 0\}) \leq n+1\}.$$

Both classes, $\mathcal{C}_1^n$ *and* $\mathcal{C}_2^n$, *cannot be learned* (*in the* Ex-*sense*) *by any* (*even non-recursive*) *learner using at most* n *mind changes. Thus,* $\mathcal{C}_1^n, \mathcal{C}_2^n \notin \mathrm{BC}_n$. *On the other hand, both classes,* $\mathcal{C}_1^n$ *and* $\mathcal{C}_2^n$, *are in* Ex_{n+1}.

4   Confidence

The notion of confidence was defined by Osherson, Stob and Weinstein [17] for set-learners. We can define confidence for function-learners in the following analogous way.

Definition 4 An Ex-learner is called *confident* if it converges on every function. (This is in general not the same as only requiring convergence on all recursive functions, see Ambainis, Freivalds and Smith [2], and Sharma, Stephan and Ventsov [18].) A BC-learner is called *confident* if it semantically converges on every function. We denote by ConfEx the family of classes that are learnable by a recursive and confident Ex-learner and by ConfBC the family of classes that are learnable by a recursive and confident BC-learner.

Ambainis, Jain and Sharma [1] showed that a class is confidently Ex-learnable if and only if it can be Ex-learned with a countable ordinal number of mind

changes. In particular, every class that is Ex-learnable with a constant number of mind changes is also confidently Ex-learnable. The next result is the corresponding one for BC: Every class BC_n is in ConfBC, indeed one even has analogous to the Ex_n-case that the learner makes at most n semantic mind changes on every function f. It needs a new proof technique since the semantic mind changes cannot be directly detected and counted down as in the case of Ex-learning. While one can trivially enforce that an Ex_n-learner makes at most n mind changes on any input function, also on functions not intended to be learned, the corresponding result for BC_n-learners is more involved.

Theorem 5 *Let* $n \in \mathbb{N}$. *Every* BC_n*-learner* M *can be translated into an at least as powerful* BC_n*-learner making at most* n *mind changes on every input function. In particular,* $\bigcup_{n \in \mathbb{N}} BC_n$ *is included in* ConfBC.

Proof. Let M be a BC_n-learner. We transform M into a BC-learner M' that learns at least the functions that M learns and makes no more than n mind changes on any input function. In order to do this, we consider the concept of seeds. Not all finite strings qualify as a seed.

Within this proof, $f(0)f(1)\ldots f(m)$ is a seed if and only if

- $M(f(0)f(1)\ldots f(m)) \in \mathbb{N}$ and
- $\varphi_{M(f(0)f(1)\ldots f(m))}(x)$ is defined and equal to $f(x)$, for $x = 0, 1, \ldots, m$.

We define the function F^α associated with a seed $\alpha = f(0)f(1)\ldots f(m)$ as follows:

$$
F^\alpha(x) = \begin{cases} f(x) & \text{if } x \leq m; \\ \varphi_{M(F^\alpha[x])}(x) & \text{if } x > m \text{ and } F^\alpha(y) \downarrow \text{ for all } y < x; \\ \uparrow & \text{otherwise.} \end{cases}
$$

Note that a program for F^α can be found effectively from α.

We say that two seeds $\alpha = f(0)f(1)\ldots f(m)$ and $\alpha' = g(0)g(1)\ldots g(m')$, where $m' \geq m$, are equivalent if and only if for all $x \leq m'$, $F^\alpha(x) = g(x)$. Note that the equivalence relation of seeds is recursively enumerable and if α and α' are equivalent then $\alpha \subseteq \alpha'$ or $\alpha' \subseteq \alpha$. Furthermore, if seeds α and α' are equivalent, then for every seed α'' such that $\alpha \subseteq \alpha'' \subseteq \alpha'$, α'' is equivalent to α.

Let $\alpha'_0, \alpha'_1, \ldots$ be a 1–1 recursive enumeration of all the seeds. Let $\alpha_0, \alpha_1, \ldots$ be obtained from $\alpha'_0, \alpha'_1, \ldots$, by suppressing all α'_i such that for some $j < i$, $\alpha'_i \subseteq \alpha'_j$. Thus for any function g, any subsequence of α_i's, which are also prefixes of g, forms a monotonic sequence.

Now the new learning algorithm M' does the following on input $g[r]$. If no $\alpha_m \subseteq g[r]$, for $m \leq r$, then $M'(g[r])$ outputs ?. Otherwise, M' on $g[r]$ outputs a program for the function F^{α_m}, for the largest $m \leq r$ such that (I) $\alpha_m \subseteq g[r]$ and (II) it can be verified in r steps that $\{\alpha_i : i < m, \alpha_i \subseteq g[r]\}$, form at most n equivalence classes.

It is now easy to verify that (a) M' on any function makes at most n semantic mind changes and (b) M' BC-learns any function BC_n-learned by M. This proves the theorem. $\square$

Definition 6 A recursive learner M learns a class $\mathcal{C}$ with the ordinal bound μ for the number of hypotheses iff there is a (not necessarily recursive) function Ord that takes arbitrary ordinals as values such that

(a) $\mu > Ord(\sigma)$ for all σ with $M(\sigma) \neq ?$ and
(b) for all σ with $M(\sigma) \neq ?$, $Ord(\sigma) > Ord(\tau)$ whenever $\sigma \preceq \tau$ and $\varphi_{M(\sigma)} \neq \varphi_{M(\tau)}$.

One can easily show that for every confident learner M, one can define Ord as required in the above definition, as one can first translate M into a non-recursive learner M' making only the semantical mind changes and omitting the other ones and then applying the known result for ConfEx-learners (see for example, [1]).

Theorem 7 *A class $\mathcal{C}$ is* ConfBC-*learnable iff $\mathcal{C}$ is* BC-*learnable with an ordinal bound on the number of hypotheses.*

If one takes the seed-learner M' from Theorem 5 one can easily verify that by taking $Ord(\sigma) = n - m$, whenever $M(\sigma) \neq ?$ and m semantic mind changes have occurred after the first $\tau \preceq \sigma$ with $M(\tau) \neq ?$, one satisfies the requirements of Definition 6. Thus we have the following proposition.

Proposition 8 *Let $n \in \mathbb{N}$. A class $\mathcal{C}$ is learnable with n semantic mind changes iff $\mathcal{C}$ is learnable with $n + 1$ hypotheses.*

Let $\mu + \nu + 1$ be the first ordinal ρ for which a set of order type ρ does not have a two-colouring into red and blue such that the subset of the red elements has order type $\leq \mu$ and that of the blue elements has order type $\leq \nu$. Then one can show that ρ has a predecessor and define $\mu + \nu$ to be this predecessor. The important difference of this definition of $+$ to the standard one having $1 + \omega = \omega$ is that, whenever $\mu \leq \mu'$ and $\nu \leq \nu'$ and one of the relations is strict, that is $\mu < \mu' \vee \nu < \nu'$, then $\mu + \nu < \mu' + \nu'$. The following theorem shows that whenever one can learn two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ with μ and ν hypotheses, then one can learn their union with $\mu + \nu$ hypotheses.

Theorem 9 *Suppose that $\mathcal{C}_1$ is* BC-*learnable with μ hypotheses and $\mathcal{C}_2$ is* BC-

learnable with ν hypotheses. Then $\mathcal{C}_1 \cup \mathcal{C}_2$ is BC-learnable with $\mu + \nu$ hypotheses. In particular, whenever $\mathcal{C}_1 \in \mathrm{BC}_m$ and $\mathcal{C}_2 \in \mathrm{BC}_n$ for natural numbers m, n, then $\mathcal{C}_1 \cup \mathcal{C}_2 \in \mathrm{BC}_{m+n+1}$.

Proof. The special case follows from the first statement of Theorem 9 in combination with Proposition 8.

The proof of the first statement uses a similar trick as in Theorem 5. Suppose that M_1 BC-learns $\mathcal{C}_1$ with μ hypotheses and M_2 BC-learns $\mathcal{C}_2$ with ν hypotheses. We say that $f(0)f(1)\ldots f(s)$ is a seed if and only if there exists an $i \in \{1, 2\}$ such that,

- $M_i(f(0)f(1)\ldots f(s)) \in \mathbb{N}$ and
- $\varphi_{M_i(f(0)f(1)\ldots f(s))}(x)$ is defined and equal to $f(x)$, for $x = 0, 1, \ldots, s$.

We define the function F^α associated with a seed $\alpha = f(0)f(1)\ldots f(s)$ as follows:

$$
F^\alpha(x) = \begin{cases}
f(x) & \text{if } x \le s, \\
\varphi_{M_i(F^\alpha[x])}(x) & \text{if } x > s,\ F^\alpha(y) \downarrow \text{ for all } y < x, \text{ and the number } i \in \{1, 2\} \text{ is the first element found, if any, in some standard search (which depends only on the sequence } F^\alpha[x]) \text{ such that } M_i(F^\alpha[x]) \downarrow \text{ and } F^\alpha(y) = \varphi_{M_i(F^\alpha[x])}(y) \text{ for all } y < x, \\
\uparrow & \text{otherwise.}
\end{cases}
$$

Note that a program for F^α can be found effectively from α.

We say that two seeds $\alpha = f(0)f(1)\ldots f(m)$ and $\alpha' = g(0)g(1)\ldots g(m')$, where $m' \ge m$, are equivalent if and only if for all $x \le m'$, $F^\alpha(x) = g(x)$. Note that the equivalence relation of seeds is recursively enumerable and if α and α' are equivalent then $\alpha \subseteq \alpha'$ or $\alpha' \subseteq \alpha$. Furthermore, if seeds α and α' are equivalent, then for every seed α'' such that $\alpha \subseteq \alpha'' \subseteq \alpha'$, α'' is equivalent to α.

Let $\alpha'_0, \alpha'_1, \ldots$ be a 1–1 recursive enumeration of all the seeds. Let $\alpha_0, \alpha_1, \ldots$ be obtained from $\alpha'_0, \alpha'_1, \ldots$, by suppressing all α'_i such that for some $j < i$, $\alpha'_i \subseteq \alpha'_j$. (Thus for any function g, any subsequence of α_i's, which are also prefixes of g, form a monotonic sequence).

Now the new learning algorithm M' does the following on input $g[r]$. If no $\alpha_m \subseteq g[r]$, for $m \le r$, then $M'(g[r])$ outputs ?. Otherwise, M' on $g[r]$ outputs a program for F^{α_s} for the largest $s \le r$ such that $\alpha_s \subseteq g[r]$.

It is easy to verify that M' BC-learns g, if M_1 BC-learns g or M_2 BC-learns g with μ or ν hypotheses, respectively. We now show that M' learns using ordinal bound $\mu + \nu$ for the number of hypothesis.

For ease of notation, we make the convention that $\varphi_?$ does not extend α.

For a seed α, let $d(\alpha)$ be a function such that $d(\alpha) = 1$, if $\varphi_{M_i(\alpha)} \supseteq \alpha$ for both $i \in \{1, 2\}$. $d(\alpha) = 0$ otherwise. Define $Ord(\alpha) = (Ord_1(\alpha) + Ord_2(\alpha)) + d(\alpha)$, where Ord_1, Ord_2 are the ordinal counters for M_1 and M_2 respectively (here we take $Ord_1(\alpha) = \mu$ if $M_1(\alpha) =?$, and $Ord_2(\alpha) = \nu$ if $M_2(\alpha) =?$).

To show the bound on number of hypothesis used by M' it suffices to show that for any two seeds α, α', if α and α' are not equivalent and $\alpha \subseteq \alpha'$, then $Ord(\alpha) > Ord(\alpha')$.

We consider two cases.

Case 1: Both $\varphi_{M_1(\alpha)}$ and $\varphi_{M_2(\alpha)}$ extend α.

In this case clearly, for some $i \in \{1, 2\}$, either M_i made a mind change between α and α' or $\varphi_{M_i(\alpha)}$ does not extend α' (otherwise, we will have that α is equivalent to α'). Thus, $Ord(\alpha') < Ord(\alpha)$.

Case 2: $\varphi_{M_i(\alpha)}$ extends α for exactly one $i \in \{1, 2\}$.

Without loss of generality assume $i = 1$, that is: $\varphi_{M_1(\alpha)}$ extends α.

Case 2.1: M_1 changes hypothesis between α and α'.

Now either M_2 changes hypothesis between α and α', or $\varphi_{M_2(\alpha')}$ does not extend α'. In both cases we have $Ord(\alpha') < Ord(\alpha)$.

Case 2.2: M_1 does not change hypothesis between α and α' and $\varphi_{M_1(\alpha)}$ does not extend α'.

In this case, since α' is a seed, $\varphi_{M_2(\alpha')}$ must extend α'. Thus M_2 must have changed hypothesis between α and α'. It follows that $Ord(\alpha') < Ord(\alpha)$.

Case 2.3: M_1 does not change hypothesis between α and α' and $\varphi_{M_1(\alpha)}$ extends α'.

In this case M_2 must change hypothesis at least once between α and α' (otherwise we would have that α and α' are equivalent).

Case 2.3.1: M_2 changes hypothesis at least twice between α and α'.

In this case clearly, $Ord(\alpha') < Ord(\alpha)$.

Case 2.3.2: M_2 changes hypothesis exactly once between α and α'.

If $\varphi_{M_2(\alpha')}$ extends α' then we would have that α is equivalent to α'. Thus, $\varphi_{M_2(\alpha')}$ does not extend α'. It follows that $Ord(\alpha') < Ord(\alpha)$.

From the above case analysis, we get that $Ord(\alpha') < Ord(\alpha)$. Also, $\mu + \nu > Ord(\alpha)$, whenever at least one of $M_i(\alpha) \neq ?$. This proves the theorem. $\square$

Note that the simulation in Theorem 9 is optimal, for μ and ν being natural numbers, as any class $\mathcal{C} \in \mathrm{Ex}_{m+n+1}$ can be split into two classes $\mathcal{C}_1, \mathcal{C}_2$ such that $\mathcal{C}_1 \cup \mathcal{C}_2 = \mathcal{C}$, $\mathcal{C}_1 \in \mathrm{Ex}_m$ and $\mathcal{C}_2 \in \mathrm{Ex}_n$. However, we have by Theorem 14 below that $\mathrm{Ex}_{m+n+1} \not\subseteq \mathrm{BC}_{m+n}$.

Blum and Blum [6] showed that Ex is not closed under finite unions. That the same holds for BC was proved by Smith [19]. In contrast to this result, the confident version of BC is closed under finite unions, as is the confident version of Ex [1,18]. This is obtained as a direct corollary of the Theorems 7 and 9.

Corollary 10 ConfBC *is closed under finite unions.*

Note 1 *Recall the notion of team learning from Section 2. The previous result can be seen as a result on team learning: In the proof of Theorem 9 we showed that two confident BC-learners can be replaced by one. By induction we see that a finite team of confident BC-learners can be replaced by one confident learner which BC-learns all the functions which are BC-learned by at least one machine in the team.*

The below theorem shows that the inclusion in Theorem 5 is strict. It should be noted that one can generalize it even to stating that there is a class in ConfEx which cannot be learned with α hypotheses where α is any fixed recursive ordinal. The diagonalizing class $\mathcal{D}$ is obtained by considering the nonincreasing functions with respect to a recursive well-ordering on $\mathbb{N}$ of order type $\alpha + 1$.

Theorem 11 ConfEx *is not contained in* $\bigcup_{n\in\mathbb{N}} \mathrm{BC}_n$.

Proof. Let $\mathcal{D}$ be the class of all nonincreasing functions. It follows from Proposition 3 that $\mathcal{D} \notin \mathrm{BC}_n$ for any n. On the other hand, $\mathcal{D} \in \mathrm{ConfEx}$: Since any $f \in \mathcal{D}$ can step down at most $f(0)$ times, we can learn $\mathcal{D}$ by a confident learner that on any input σ makes sure that no more than $\sigma(0)$ syntactic changes have been made. $\square$

5   Anomalous hypotheses

In this section we discuss learning with a finite number of anomalies. In both the Ex and the BC case it is known that allowing final hypotheses that are defective at a finite number of inputs, either by being undefined or by giving the wrong answer, increases the number of classes that can be effectively learned. For partial functions η and ψ, let $\eta =^* \psi$ denote that for almost every x, $\eta(x) = \psi(x)$. (As usual, we take $\eta(x) = \psi(x)$ to mean that if one of $\eta(x),\psi(x)$ is undefined, then the other one is too.) Similarly, $\eta =^n \psi$ means that $\eta(x) = \psi(x)$ for all x, with the possibility of at most n exceptions. Now Ex^* and Ex^n are defined similarly to Ex except that instead of requiring the final hypothesis k to be a program for f, we require $\varphi_k =^* f$ and $\varphi_k =^n f$ respectively. Similarly for BC^* and BC^n. For example M BC^n-learns a function f if for almost every k, $\varphi_{M(f[k])} =^n f$. We define BC^n_m as follows.

Definition 12 Let $n, m \in \mathbb{N}$. A learner M BC^n_m-*learns* a function f whenever M BC^n-learns f with at most m semantic mind changes. BC^n_m denotes the family of classes that can be recursively BC^n_m-learned.

We note that there is at least one other (nonequivalent) way of defining BC^n_m, where one also counts the semantic mind changes modulo finite differences. That is, one considers a mind change to have taken place by M at $f[k+1]$, if $M(f[k]) \neq ?$ and $\varphi_{M(f[k])} \neq^n \varphi_{M(f[k+1])}$. However, this definition is mathematically less elegant. For example the relation "$=^n$" is not transitive and so it might happen that $\varphi_{M(f[k])} =^n \varphi_{M(f[k+1])}$ and $\varphi_{M(f[k+1])} =^n \varphi_{M(f[k+2])}$ while $\varphi_{M(f[k])} \neq^n \varphi_{M(f[k+2])}$. Furthermore, there would be nontrivial collapses like $\text{BC}^1_0 = \text{BC}^2_0$ with respect to the alternative definition.

Steel [15] noticed that $\text{Ex}^* \subseteq \text{BC}$. The next result shows that a smaller bound on the number of mind changes cannot be compensated by permitting errors and using semantic instead of syntactic mind changes. Note that the result provides the omitted proofs of parts (c) and (d) of Theorem 2.

The following proposition can be proved using easy diagonalizations, similar to the ones found in [8]. We leave the details to the reader.

Proposition 13 (Based on [8]) *Let $n \in \mathbb{N}$. Let $\mathcal{C}^n_1$ be as in Proposition 3. Then $\mathcal{C}^n_1 \notin \text{BC}^*_n$.*

Theorem 14 *For every $n \in \mathbb{N}$ it holds that $\text{Ex}_{n+1} \not\subseteq \text{BC}^*_n$. Furthermore, $\text{Ex} \not\subseteq \bigcup_{n \in \mathbb{N}} \text{BC}^*_n$.*

Proof. The family $\mathcal{C}^n_1$ from Proposition 3 witnesses that $\text{Ex}_{n+1} \not\subseteq \text{BC}^*_n$ (by Proposition 3 and Proposition 13). Let $\mathcal{C} = \bigcup_{n \in \mathbb{N}} \mathcal{C}^n_1$. Clearly, $\mathcal{C} \in \text{Ex}$. However, $\mathcal{C} \notin \bigcup_{n \in \mathbb{N}} \text{BC}^*_n$ by Proposition 13. $\square$

Blum and Blum [6, page 152] stated that $\text{Ex}^* \not\subseteq \text{Ex}$. Bārzdins [4] proved that $\text{BC} \not\subseteq \text{Ex}$. Case and Smith [8, Theorem 2.3] proved that the class $\mathcal{S}^1 = \{f : \varphi_{f(0)} =^1 f\}$ is in $\text{Ex}^1 - \text{Ex}$. Clearly $\mathcal{S}^1 \in \text{BC}_1$ so it follows immediately that $\text{BC}_1 \not\subseteq \text{Ex}$. Case and Smith and Harrington [8, Theorem 3.1] proved that the class $\{f : (\forall^\infty x)\, [\varphi_{f(x)} = f]\}$ is in $\text{BC} - \text{Ex}^*$. From this proof actually follows the stronger statement that the smaller class

$$\mathcal{X} = \{f : (\exists n)\, (\forall i)\, [i \leq n \rightarrow \varphi_{f(i)} = \varphi_{f(0)} \wedge i > n \rightarrow \varphi_{f(i)} = f]\}$$

is in $\text{BC} - \text{Ex}^*$. Since $\mathcal{X}$ is clearly in BC_1 this gives us the following result.

Theorem 15 BC_1 *is not included in* Ex^*.

Theorem 15 will be improved in Theorem 19.

The following result shows that in the BC model there is a tradeoff between mind changes and anomalies. Note that this is different in the Ex model where there is no such tradeoff. Namely, Case and Smith [8] proved that Ex_0^1 is not contained in Ex. Tradeoff results for a different notion of mind change in the context of vacillatory function learning were studied in Case, Jain and Sharma [7].

Theorem 16 *Let* $n, m \in \mathbb{N}$. BC_m^n *is included in* $\text{BC}_{n(m+1)+m}$. *For* $n > 0$ *the inclusion is strict. Furthermore, the bound* $n(m+1) + m$ *is optimal.*

Proof. Proof of the inclusion: Let M be a BC_m^n-learner. We will try to overcome anomalies by hard-wiring bits of the input data, in such a way as to make the least possible number of semantic changes. Hard-wiring all values of the input data can already make this number recursively unbounded when the first hypotheses of M are wrong, so we have to be more careful. Since we know that the "final" hypotheses of M are faulted at at most n places, we never patch more than n inputs. That is, we transform every hypothesis $M(\sigma)$ into an hypothesis $M'(\sigma)$ that implements the following algorithm. Compute the longest $\tau \preceq \sigma$ such that there are at most n places $x \in \text{dom}(\tau)$ with either $\varphi_{M(\tau),s}(x) \uparrow$ or $\varphi_{M(\tau),s}(x) \downarrow \neq \tau(x)$, where $s = |\sigma|$. Then let

$$\varphi_{M'(\sigma)}(x) = \begin{cases} \tau(x) & \text{if } x \in \text{dom}(\tau), \\ \varphi_{M(\tau)}(x) & \text{if } x \notin \text{dom}(\tau). \end{cases}$$

So the algorithm has two ingredients: delaying and patching. It is easy to verify that every mind change is either caused by patching some x with $\tau(x)$ that has been incorrect before or by following an original mind change of M. Between two (delayed) semantic mind changes of M there are at most n places at which M' causes a mind change by patching one input. So patching may induce up to n mind changes between two delayed ones plus n mind changes

before the first (delayed) mind change of M and n mind changes after the last (delayed) mind change of M. Together with the up to m original mind changes of M this gives altogether at most $n(m + 1) + m$ mind changes.

Furthermore the last hypothesis of M agrees with the function to be learned on all but up to n places. These at most n places are repaired by patching. So whenever M BC_m^n-learns a function f, M' $\mathrm{BC}_{n(m+1)+m}$-learns the same function f.

Proof of the strictness of the inclusion when $n > 0$: This follows immediately from Theorem 14.

Proof of the optimality of the bound: We prove that Ex_m^n is not included in $\mathrm{BC}_{n(m+1)+m-1}$. Consider the class $\mathcal{S}$ of functions that are zero at all but up to $n(m + 1) + m$ inputs. Then $\mathcal{S} \notin \mathrm{BC}_{n(m+1)+m-1}$ by Proposition 3. On the other hand, $\mathcal{S} \in \mathrm{Ex}_m^n$ because an Ex_m^n-learner can output its $(j + 1)$-th guess after having seen $j(n + 1)$ nonzero values in the input function (where the guess is the zero-extension of the function seen so far; note that the $(j + 1)$-th guess would make at most n errors as long as there are $\leq (j + 1)(n + 1) - 1$ non-zero values in the input function). In this way, with m mind changes the Ex_m^n-learner can handle upto $(n + 1)(m + 1) - 1$ nonzero values in the input function. Hence $\mathcal{S} \in \mathrm{Ex}_m^n$.   □

Next we consider learning by team of learners (see Section 2). First we prove that $\mathrm{BC}_n \subseteq [1, n + 1]\mathrm{Ex}$ and that $\mathrm{BC}_n \nsubseteq [1, n]\mathrm{Ex}^*$.

Theorem 17 BC_n *is strictly included in* $[1, n + 1]\mathrm{Ex}$ *for every* $n \in \mathbb{N}$.

Proof. Let M witness that $\mathcal{S} \in \mathrm{BC}_n$ and let $\mathcal{S}_k$ be the subclass of those functions in $\mathcal{S}$ where M makes exactly k semantic mind changes. Clearly $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n$.

For each class $\mathcal{S}_k$ there is an Ex-learner M_k: The machine M_k searches for the least tuple $(\sigma_0, x_0, \sigma_1, x_1, \ldots, \sigma_{k-1}, x_{k-1}, \sigma_k)$ that is a candidate for witnessing k semantic mind changes. M_k computes at every $f[m] \preceq f$ an m-th approximation to this tuple and outputs $M(\sigma_k)$ for this approximation.

The search conditions for this tuple to witness the k semantic mind changes are the following three.

- $\sigma_0 \prec \sigma_1 \prec \ldots \prec \sigma_k \preceq f$ where f is the function to be learned,
- $M(\sigma_h) \neq ?$ for every $h \leq k$,
- $\varphi_{M(\sigma_h)}(x_h) \neq \varphi_{M(\sigma_{h+1})}(x_h)$ (i.e. either exactly one of the values is undefined or both are defined but different) for every $h < k$.

Note that for the learner M_0 the first and the third condition are void so that the only search condition is to find some $\sigma_0 \preceq f$ with $M(\sigma_0) \neq ?$. The last condition can only be verified in the limit, so it might happen that either a correct tuple needs some time until it qualifies or that some incorrect tuple is considered to be a candidate until it is disqualified.

If $f \in \mathcal{S}_k$ then there exist such tuples and M_k converges to the least one of them. It follows that $M_k(f[m])$ converges to $M(\sigma_k)$ for the σ_k of this least tuple. The candidates for the mind changes are then correct. So M makes k mind changes before seeing σ_k and no mind change later. So $M(\sigma_k)$ is indeed a program for f and M_k is an Ex-learner for $\mathcal{S}_k$. It follows that the team $M_0, M_1, \ldots, M_n$ infers the whole class $\mathcal{S}$ with respect to the criterion $[1, n + 1]$Ex.

The strictness of the inclusion follows from (the proof of) Theorem 11 showing that the class $\mathcal{D}$ is in ConfEx and thus in $[1, n + 1]$Ex, but not in BC_n. □

Theorem 18 BC_n *is not included in* $[1, n]$Ex* *for every* $n \in \mathbb{N}$.

Proof. Let $\mathcal{S}^1 = \{f : \varphi_{f(0)} =^1 f\}$. (See also the discussion preceding Theorem 15.) Let $\mathcal{S}^n = \{f_1 \oplus \cdots \oplus f_n : f_i \in S^1\}$. Here, given $f_1, \ldots, f_n$, the function $f = f_1 \oplus \cdots \oplus f_n$ is defined by $f(a \cdot n + b) = f_{b+1}(a)$ where $a \in \{0, 1, 2, \ldots\}$ and $b \in \{0, 1, \ldots, n - 1\}$. It follows from Kummer and Stephan [13, Theorem 8.2] that $\mathcal{S}^n \notin [1, n]$Ex, whereas it is easy to see (by combining the codes of the f_i) that $\mathcal{S}^n \in \text{Ex}_0^n \subseteq BC_n$. To obtain a result for $[1, n]$Ex*, define the cylindrification $\text{Cyl}(\mathcal{S}^n) = \{f : (\exists g \in \mathcal{S}^n)(\forall x, y)[f(\langle x, y \rangle) = g(x)]\}$. Since for any class $\mathcal{A}$ it holds that $\text{Cyl}(\mathcal{A}) \in [1, n]$Ex if and only if $\text{Cyl}(\mathcal{A}) \in [1, n]$Ex*, and $\text{Cyl}(\mathcal{A}) \in [1, n]$Ex implies $\mathcal{A} \in [1, n]$Ex, it follows that $\text{Cyl}(\mathcal{S}^n) \notin [1, n]$Ex*. However, the BC_n-algorithm for $\mathcal{S}^n$ easily transfers to $\text{Cyl}(\mathcal{S}^n)$. □

Case and Smith [8] introduced the notion of OEx*-learning. In this criteria, the learner outputs finitely many indices such that at least one of these indices computes a finite variant of f. Case and Smith [8] proved that neither of the classes BC and OEx* is included in the other. The next result improves on one of these noninclusions by showing that BC_1 is not contained in OEx*. Note that since Ex$^* \subseteq$ OEx*, this also improves Theorem 15.

Theorem 19 BC_1 *is not contained in* OEx*.

Proof. The class $\text{Cyl}(\mathcal{S}^1)$, the cylindrification of the class $\mathcal{S}^1$ (see the proof of Theorem 18), is in BC_1. Suppose for a contradiction that $\text{Cyl}(\mathcal{S}^1)$ is in OEx*, and that M is a total OEx*-learner for it.

Now a family of partial functions ψ_e is constructed, using for each ψ_e a marker m_e; after each step s the domain of ψ_e is $\{0, 1, \ldots, s\} - \{m_{e,s}\}$ where $m_{e,s}$

is the marker position after step s. The intention of construction for ψ_e is to show that there is a function $f_e \in \mathrm{Cyl}(S^1)$ that is an extension of the function $\langle x, y \rangle \mapsto \psi_e(x)$ and which is not OEx*-learned by M.

- In step 1 define $\psi_e(0) = e$ and place m_e on the position 1, that is, let $m_{e,1} = 1$.
- In step $s + 1$, $s \geq 1$, for all $a, b \leq s + 1$ define the strings $\sigma_{a,b}$ such that the domain of $\sigma_{a,b}$ is the longest interval $\{0, 1, \ldots, u_b\}$ where all pairs $\langle x, y \rangle \leq u_b$ satisfy $x < b$ and

$$\sigma_{a,b}(\langle x, y \rangle) = \begin{cases} \psi_e(x) & \text{if } x \neq m_{e,s}, \\ a & \text{if } x = m_{e,s}. \end{cases}$$

- Then check whether there is a value $a \leq s + 1$ such that M outputs on some input σ with $\sigma_{a,m_{e,s}} \prec \sigma \preceq \sigma_{a,s+1}$ a new guess which has not been seen before.
- If so, then let $\psi_e(m_{e,s}) = a$ and move the marker to the next still undefined position of ψ_e: $m_{e,s+1} = s + 1$.
- If not, then let $\psi_e(s + 1) = 0$ and let the marker stay where it is: $m_{e,s+1} = m_{e,s}$.

If the marker moves infinitely often then ψ_e is total; otherwise ψ_e is defined at all inputs except the end-position $m_{e,\infty}$ of the marker m_e. By the Recursion Theorem there is an index e with $\varphi_e = \psi_e$; fix such index e and note that all extensions of ψ_e are in S^1.

If the marker m_e moves infinitely often, then ψ_e is total and the function f_e given by $f_e(\langle x, y \rangle) = \psi_e(x)$ is in $\mathrm{Cyl}(S^1)$. It follows from the construction that M outputs infinitely many different guesses on f_e. So M does not OEx*-learn f_e which gives the desired contradiction for this case.

So it remains to look at the scenario when m_e moves only finitely often and remains on the end-position $m_{e,\infty}$. Now define the functions

$$f_{e,a}(\langle x, y \rangle) = \begin{cases} \psi_e(x) & \text{if } x \neq m_{e,\infty}, \\ a & \text{if } x = m_{e,\infty}. \end{cases}$$

M shows on all these functions the same behaviour in the sense that it outputs the same finite set E of indices — since otherwise there would be an a permitting a new output outside E and the marker would move again. Furthermore all functions $f_{e,a}$ are in $\mathrm{Cyl}(S^1)$ and they differ on infinitely many values. So only finitely many of these functions have a program in E that computes them at almost all places. Thus, one can choose a such that no program in E computes $f_e = f_{e,a}$ with at most finitely many errors.

15

So in both cases there is a function $f_e \in \mathrm{Cyl}(S^1)$ which is not learned by M under the criterion OEx^* and it follows that $\mathrm{Cyl}(S^1)$ is a witness for the non-inclusion $\mathrm{BC}_1 \not\subseteq \mathrm{OEx}^*$.   □

Recall the notion of confidence from Definition 4. A class is in ConfEx^* if it is Ex^*-learned by a learner that converges on every function. Since every Ex_m^*-learner can easily be converted into a ConfEx^*-learner we have the inclusion $[1, n]\mathrm{Ex}_m^* \subseteq [1, n]\mathrm{ConfEx}^*$. Furthermore, every ConfEx^*-learner outputs on every function only finitely many indices, so a team of n ConfEx^*-learners in total also outputs on every function finitely many indices. Thus it follows that $[1, n]\mathrm{ConfEx}^* \subseteq \mathrm{OEx}^*$. As a consequence, BC_1 is not contained in any of the just mentioned criteria. Smith [19, Theorem 3.8] proved that $\mathrm{BC} \not\subseteq \bigcup_{n \in \mathbb{N}} [1, n]\mathrm{Ex}^*$. This may be compared to the following corollary.

Corollary 20 *For every $n, m \in \mathbb{N}$, BC_1 is neither a subclass of $\bigcup_{n,m \in \mathbb{N}} [1, n]\mathrm{Ex}_m^*$ nor a subclass of $\bigcup_{n \in \mathbb{N}} [1, n]\mathrm{ConfEx}^*$.*

Note that it makes sense to consider teams in the case of learning with finitely many errors since teams of ConfEx^*-learners have more power than single ConfEx^*-learners: The class containing the functions that are zero almost everywhere and the functions that are self-describing is learnable by a $[1, 2]\mathrm{Ex}_0^*$ team but not by a single Ex^*-learner [8, Theorem 2.13].

We also remark that the proof of Theorem 11 shows that in fact ConfEx is not included in $\bigcup_{n \in \mathbb{N}} \mathrm{BC}_n^*$.

The results presented in this paper do not resolve all the relationship between different BC_n^m criteria, which is an open problem. Similarly, for the case involving team of learners. In this respect note that since the classes $[a, b]\mathrm{BC}_0$ and $[a, b]\mathrm{Ex}_0$ are the same and the exact relation between the classes $[a, b]\mathrm{Ex}_0$ is still unknown, the same holds for the classes $[a, b]\mathrm{BC}_n$. Nevertheless many results have already been obtained for the inclusion relation of $[a, b]\mathrm{Ex}_0$. For a list of references see [11, p 219].

6   Grammar induction

In this section we make some remarks on grammatical inference. In the previous sections we have been concerned with the inductive inference of computable functions. Here we consider the more general paradigm of learning recursively enumerable sets, or, when we think of the code of a recursively enumerable set as a grammar generating the set, the learning of grammars from pieces of text. The set learning analogs of the models Ex and BC that

we studied in the previous sections are defined as follows (we use the notation of [11]): Let W_e denote the domain of φ_e, i.e. the set accepted by the e-th program in the standard acceptable numbering φ.

Definition 21 Let L be a recursively enumerable set. A *text* t for L is a (not necessarily recursive) mapping from $\mathbb{N}$ to $L \cup \{\#\}$ such that all elements of L appear in the range of t (# may or may not appear in the range of t; The usage of # is to allow texts for empty language).

The initial segment of length n of t is denoted by $t[n]$. A learner M TxtEx-*learns* L if for every text t for L, $\lim_{n\to\infty} M(t[n]) = e$ exists and $W_e = L$. M TxtBC-*learns* L if for every text t for L, $W_{M(t[n])} = L$ for almost every n. A machine M TxtBC$_n$-*learns* L (or: M TxtBC-learns L with at most n *semantic mind changes*) if M TxtBC-learns L such that the cardinality of the set $\{n : M(t[n]) \neq ? \wedge W_{M(t[n])} \neq W_{M(t[n+1])}\}$ is at most n. A class $\mathcal{L}$ of recursively enumerable sets is in TxtEx [TxtBC, TxtBC$_n$] when there is a recursive learner that TxtEx-learns [TxtBC-learns, TxtBC$_n$-learns] every $L \in \mathcal{L}$. Variants of these classes, such as the analog TxtBC^n of BC^n, are defined in the obvious way.

The definition of confidence for language-learners is as follows:

Definition 22 A TxtEx-learner is *confident* if it converges on every text. A TxtBC-learner is *confident* if it TxtBC-converges on every text. We denote by ConfTxtBC the classes that are TxtBC-learnable by a confident learner.

First we note that a negative result on function learning immediately yields a corresponding negative result for language learning, since the latter is a more general setting. (We can embed the first into the second by interpreting the graph of a recursive function as a simple kind of recursively enumerable set.) Thus, the Theorems 2, 11, 14, 15, 18 and 19 all hold for the corresponding models of language learning. The following simple result shows that Theorem 16 does not transfer.

Theorem 23 (See [11, page 145, 147]) TxtBC$_0^1$ *is not contained in* TxtBC, *as witnessed by the class* $\{W_e : W_e =^1 \mathbb{N}\}$.

Proof. Consider the class $\mathcal{X} = \{W_e : W_e =^1 \mathbb{N}\}$. $\mathcal{X} \in$ TxtBC$_0^1$ since it is learned by the learner that always outputs a code for $\mathbb{N}$. On the other hand, it follows from Angluin's characterization of learnability without errors [11, Theorem 3.26] that $\mathcal{X}$ is not learnable by any learner (even when nonrecursive learners are allowed). In particular $\mathcal{X}$ is not TxtBC-learnable. $\square$

Finally, it is easy to see that the idea for the proof of Theorem 17 can be used to show that this result also holds for language learning. We now consider Corollary 10. We want to show that Corollary 10 does not hold for language

learning. For this we use the following result, which is interesting in itself. First a definition:

Definition 24 Let $\mathcal{L}$ be a collection of recursively enumerable sets.

(I) (Angluin [3]) $\mathcal{L}$ has *finite thickness* if for every finite $D \neq \emptyset$ the collection $\{L \in \mathcal{L} : D \subseteq L\}$ is finite.

(II) $\mathcal{L}$ is *finite-to-1 enumerable* if there is a recursive function f such that $\mathcal{L} = \{W_{f(i)} : i \in \mathbb{N}\}$ and for every member $L \in \mathcal{L}$ there are at most finitely many i such that $L = W_{f(i)}$. (Note that this finite number may depend on L.) Similarly, $\mathcal{L}$ is *1-1-enumerable* if it has an enumeration in which every set has only one code.

Theorem 25 *There exists a uniformly recursively enumerable collection $\mathcal{L}$ that has finite thickness and that is not in* TxtBC.

Proof. The proof is an adaptation of the proof of Theorem 3.1 in Terwijn [22] (which showed that there is a 1-1-enumerable learnable collection of recursive sets that is not in TxtBC). The collection $\mathcal{L}$ contains for every e a subclass $\mathcal{L}_e$ such that the e-th partial recursive function φ_e does not TxtBC-learn $\mathcal{L}_e$. To separate the strategies for different e we let the elements of $\mathcal{L}_e$ be subsets of $\mathbb{N}^{[e]} = \{\langle e, x \rangle : x \in \mathbb{N}\}$.

The classes $\mathcal{L}_e$ are uniformly enumerated as follows. $\mathcal{L}_e$ will contain $L_{e,0} = \mathbb{N}^{[e]}$, a certain diagonal set $L_{e,1}$ and sets $L_{e,j}$, $j > 1$, such that at least one of the following cases holds:

- φ_e does not TxtBC-learn $L_{e,1}$. Furthermore, every $L_{e,i}$, $i > 1$, will be either empty or equal to $L_{e,0}$.
- φ_e does not TxtBC-learn a $L_{e,j}$ with $j > 1$. Furthermore, every $L_{e,i}$, with $1 < i < j$, will equal $L_{e,0}$ and all $L_{e,i}$ with $i > j$ will be empty.

The construction of $\mathcal{L}_e$ is now as follows. We use auxiliary variables $x_{e,j}$ and σ_j.

Initialization: Let σ_0 be the empty string, $L_{e,0} = \mathbb{N}^{[e]}$, $L_{e,j} = \emptyset$ for all $j > 0$. In subsequent stages we may add elements to these sets. Go to stage 1. Stage j. For all i with $1 < i < j$, let $L_{e,i} = L_{e,0}$ and $L_{e,j+1} = L_{e,0} - \{x_{e,1}, x_{e,2}, \dots, x_{e,j-1}\}$. Search for a number $x_{e,j}$ in $L_{e,j+1}$ and an extension σ_j of σ_{j-1} such that the range of σ_j contains only elements from $\mathbb{N}^{[e]} - \{x_{e,1}, \dots, x_{e,j}\}$, $\varphi_e(\sigma_j)$ is defined and the set $W_{\varphi_e(\sigma_j)}$ generated by it contains $x_{e,j}$. If these are found, add the range of σ_j to $L_{e,1}$, and go to Stage $j + 1$.

This completes the construction of the $\mathcal{L}_e$. Now there are two possibilities:

- The construction of $\mathcal{L}_e$ is completed at every stage j. Then the union of all the σ_j constitute a text for $L_{e,1}$, but φ_e infinitely often outputs an hypothesis

18

that contains a non-element of $L_{e,1}$. Hence φ_e does not TxtBC-learn $L_{e,1}$.

- Stage j in the construction is not completed for some j. In this case $x_{e,j}$ is not found and the learner φ does not overgeneralize on any text for $L_{e,j+1}$ starting with σ_{j-1}. Hence φ does not TxtBC-learn $L_{e,j+1}$.

Note that every $\mathcal{L}_e$ has finite thickness since it contains at most the sets $L_{e,0}$, $L_{e,1}$ and possibly some $L_{e,j}$. □

Terwijn [22, Theorem 5.3] showed that a finite-to-1 enumerable collection that has finite thickness is in TxtBC. Theorem 25 shows that the hypothesis of finite-to-1 enumerability is necessary for this result. Now we use the proof of Theorem 25 to show that the analog of Corollary 10 fails for language learning.

Theorem 26 *There are classes $\mathcal{C}_0$ and $\mathcal{C}_1$ in $\mathrm{ConfTxtBC}_1$ such that $\mathcal{C}_0 \cup \mathcal{C}_1$ is not in TxtBC. Hence neither $\mathrm{ConfTxtBC}$ nor TxtBC_n, $n \geq 1$, is closed under finite unions.*

Proof. Let $\mathcal{L}$ be the collection from the proof of Theorem 25. This collection contains for every e a set $L_{e,1}$. Let $\mathcal{C}_0$ be the collection consisting of all these $L_{e,1}$'s, plus the empty set. Clearly $\mathcal{C}_0$ is in $\mathrm{ConfTxtBC}_1$. We now prove that also $\mathcal{C}_1 = \mathcal{L} - \mathcal{C}_0$ is in $\mathrm{ConfTxtBC}_1$. Since by Theorem 25 $\mathcal{C}_0 \cup \mathcal{C}_1 = \mathcal{L}$ is not in TxtBC the theorem follows. We define a confident recursive TxtBC-learner M for $\mathcal{C}_1$. We use the notation of the proof of Theorem 25. Given a piece of text σ: If σ contains no elements, then M outputs ?. Otherwise M finds e such that σ contains elements only from $\mathbb{N}^{[e]}$. M then follows the definition of $L_{e,1}$ for $|\sigma|$ steps in order to find the first "gap" $x_{e,1}$. If $x_{e,1}$ is not found, M outputs $\mathbb{N}^{[e]}$ as a guess. If $x_{e,1}$ is found and is in the range of σ, then σ can only be a subset of $L_{e,0}$ (among languages in $\mathcal{C}_1$). Thus M can safely output a grammar for $L_{e,0} = \mathbb{N}^{[e]}$. Otherwise, let $M(\sigma)$ be the program that searches for $|\sigma|$ steps for as many gaps $x_{e,i}$ as possible. If after $|\sigma|$ steps $x_{e,1}, \ldots, x_{e,l}$ are found, $M(\sigma)$ starts to enumerate $L_{e,0} - \{x_{e,1}, \ldots, x_{e,l}\}$. If, however, in the course of this enumeration another gap $x_{e,l+1}$ is found, M knows its guess is wrong and starts to enumerate all of $L_{e,0}$. Now if there is indeed an infinite number of gaps $x_{e,i}$, then $M(\sigma)$ is always a code for $L_{e,0}$. If there is only a finite number of gaps $x_{e,1}, \ldots, x_{e,l}$, then $M(\sigma)$ is almost always a code for $L_{e,0} - \{x_{e,1}, \ldots, x_{e,l}\}$. Note that in this last case there is also at most one semantic mind change. So M is confident and it TxtBC_1-learns $\mathcal{C}_1$. □

We note without proof that, in analogy to Theorem 26, there are two classes in TxtEx_0 whose union is not in TxtEx. However, in Theorem 26 one cannot get TxtBC_0 instead of TxtBC_1 since the union of two classes in $\mathrm{ConfTxtEx}$ is in $\mathrm{ConfTxtBC}$ and every TxtEx_n-learnable class is $\mathrm{ConfTxtEx}$-learnable.

References

[1] A. Ambainis, S. Jain and A. Sharma, *Ordinal mind change complexity of language identification*, Theoretical Computer Science, 220 (1999) 323-343. Special issue on Australasian Computer Science.

[2] A. Ambainis, R. Freivalds and C. H. Smith, *Inductive Inference with Procrastination: Back to Definitions*, Fundamenta Informaticae 40 (1999) 1–16.

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

[4] J. M. Bārzdins, *Two theorems on the limiting synthesis of functions*, Theory of Algorithms and Programs (Latvian State University) 1 (1974) 82-88 (in Russian).

[5] J. M. Bārzdins and R. Freivalds, *On the prediction of general recursive functions*, Soviet Mathematics Doklady 13 (1972) 1224-1228.

[6] L. Blum and M. Blum, *Toward a mathematical theory of inductive inference*, Information and Control 28 (1975) 125-155.

[7] J. Case, S. Jain and A. Sharma, *Complexity issues for vacillatory function identification*, Information and Computation 116(2) (1995) 174–192.

[8] J. Case and C. Smith, *Comparison of identification criteria for machine inductive inference*, Theoretical Computer Science 25 (1983) 193-220.

[9] R. Freivalds and C. Smith, *On the role of procrastination in machine learning*, Information and Computation 107 (1993), 237–271.

[10] E. M. Gold, *Language identification in the limit*, Information and Control 10 (1967) 447-474.

[11] S. Jain, D. Osherson, J. S. Royer and A. Sharma, *Systems that learn, An introduction to learning theory*, second edition, MIT Press, 1999.

[12] M. J. Kearns, U. V. Vazirani, *An introduction to computational learning theory*, MIT Press, 1994.

[13] M. Kummer and F. Stephan, *On the structure of degrees of inferability*, Journal of Computer and System Sciences 52 (1996) 214–238.

[14] P. Odifreddi, *Classical Recursion Theory*, North-Holland, Amsterdam, 1989.

[15] P. Odifreddi, *Inductive inference of total functions*, in: S. B. Cooper, T. A. Slaman, S. S. Wainer (eds.), *Computability, Enumerability, Unsolvability. Directions in Recursion Theory*, London Math. Soc. Lecture Note Series 224 (1996) 259–288.

[16] D. N. Osherson, M. Stob and S. Weinstein, *Aggregating inductive expertise*, Information and Computation 70(1) (1986) 69–95.

[17] D. Osherson, M. Stob and S. Weinstein, *Systems that learn, An introduction to learning theory*, MIT Press, 1986.

[18] A. Sharma, F. Stephan, Y. Ventsov, *Generalized notions of mind change complexity*, Proceedings of the Tenth Conference on Computational Learning Theory (COLT'97), Nashville (1997) 96–108.

[19] C. Smith, *The power of pluralism for automatic program synthesis*, J. ACM 29, Vol. 4 (1982) 1144-1165.

[20] R. I. Soare, *Recursively enumerable sets and degrees*, Springer-Verlag, 1987.

[21] F. Stephan and S. A. Terwijn, *Counting extensional differences in BC-learning*, Proceedings of the 5th International Colloquium on Grammatical Inference (ICGI 2000), Springer Lecture Notes in A. I. 1891 (2000) 256–269.

[22] S. A. Terwijn, *Extensional set learning*, Proceedings of The Twelfth Annual Conference on Computational Learning Theory (COLT '99), Santa Cruz (1999) 243–248.