Learning Classes of Approximations to Non-Recursive Functions [1]

Frank Stephan [a,2], Thomas Zeugmann [b,3]

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

[b] *Medizinische Universität Lübeck, Institut für Theoretische Informatik, Wallstraße 40, 23560 Lübeck, Germany, Email: thomas@tcs.mu-luebeck.de*

Abstract

Blum and Blum (1975) showed that a class $\mathcal{B}$ of suitable recursive approximations to the halting problem K is reliably EX-learnable but left it open whether or not $\mathcal{B}$ is in NUM. By showing $\mathcal{B}$ to be not in NUM we resolve this old problem.

Moreover, variants of this problem obtained by approximating any given recursively enumerable set A instead of the halting problem K are studied. All corresponding function classes $\mathcal{U}(A)$ are still EX-inferable but may *fail* to be reliably EX-learnable, for example if A is *non-high and hypersimple.*

Blum and Blum (1975) considered only approximations to K defined by monotone complexity functions. We prove this condition to be necessary for making learnability independent of the underlying complexity measure. The class $\tilde{\mathcal{B}}$ of all recursive approximations to K generated by all total complexity functions is shown to be not even behaviorally correct learnable for a class of natural complexity measures. On the other hand, there are complexity measures such that $\tilde{\mathcal{B}}$ is EX-learnable. A similar result is obtained for all classes $\tilde{\mathcal{U}}(A)$.

For natural complexity measures, $\mathcal{B}$ is shown to be not *robustly* learnable, but again there are complexity measures such that $\mathcal{B}$ and, more generally, every class $\mathcal{U}(A)$ is robustly EX-learnable. This result extends the criticism of Jain, Smith and Wiehagen (1998), since the classes defined by artificial complexity measures turn out to be robustly learnable while those defined by natural complexity measures are not robustly learnable.

---

[1] Updated Version of: Learning Classes of Approximations to Non-Recursive Functions. Department of Informatics, Report 166, Kyushu University, Fukuoka, 1999.

[2] Supported by the Deutsche Forschungsgemeinschaft (DFG) under Heisenberg grant no. Ste 967/1–1.

[3] Supported by the Grant-in-Aid for Scientific Research in Fundamental Areas from the Japanese Ministry of Education, Science, Sports, and Culture under

1 Introduction

Though algorithmic learning of recursive functions has been intensively studied within the last three decades there is still some need to elaborate this theory further. For the purpose of motivation, let us shortly recall the basic scenario.

An algorithmic learner is fed growing initial segments of the graph of the target function f. Based on the information received, the learner computes a hypothesis on each input. The sequence of all computed hypotheses has to converge to a correct, finite and global description of the target f. We shall refer to this scenario by saying that f is *EX-learnable* (cf. Definition 2).

Clearly, what one is really interested in are powerful learning algorithms that cannot only learn one function but all functions from a given class of functions. Gold [13] provided the first such powerful learner, i.e., the *identification by enumeration* algorithm and showed that it can learn every class contained in *NUM*. Here *NUM* denotes the family of all function classes that are subsets of some recursively enumerable class of recursive functions.

There are, however, learnable classes of recursive functions which are not contained in *NUM*. The perhaps most prominent example is the class $\mathcal{SD}$ of self-describing recursive functions, i.e., of all those functions that compute a program for themselves on input 0. Clearly, $\mathcal{SD}$ is *EX*-learnable.

Since Gold's [13] pioneering paper a huge variety of learning criteria have been proposed within the framework of inductive inference of recursive functions (cf., e.g., [3,6,8,10,18,20,23,27]). By comparing these inference criteria to one another, it became popular to show separation results by using function classes with self-referential properties. On the one hand, the proof techniques developed are mathematically quite elegant. On the other hand, these separating examples may be considered to be a bit artificial, because of the use of self-describing properties. So, the search for natural examples to separate learning criteria is still an active research area. Within this work, we attack the following two main goals:

(1) What is the overall theory developed so far telling us about the learnability of "naturally defined function classes?"
(2) How behave "naturally defined function classes" with respect to criteria in the literature designed to characterize "natural learnability," i.e., notions

grant no. 10558047. Part of this work was done while visiting the Laboratoire d'Informatique Algorithmique: Fondements et Applications, Université Paris 7. This author is gratefully indebted to Maurice Nivat for providing financial support and inspiring working conditions.

of learnability not exploiting self-reference?

Clearly an answer to the first question should tell us something about the usefulness of the theory. An answer to the second problem should, in particular, provide some insight into the "appropriateness" of the proposed notions of "natural learnability." However, our knowledge concerning both questions has been severely limited. For recursively enumerable function classes almost everything is clear, i.e., their learnability has been proved with respect to many learning criteria.

A prominent example for a "natural" function class outside NUM is the set of all general recursive complexity functions of an abstract complexity measure introduced by Blum [5]. But already for this class the proposed notions of "natural learnability" seem to fail (see below).

The situation may be completely different if one looks at classes of $\{0,1\}$-valued recursive functions, since their learnability differs sometimes considerably from the inferability of arbitrary function classes (cf., e.g., [21,32]). As far as these authors are aware of, one of the very few "natural classes" of $\{0,1\}$-valued recursive functions that may be a candidate to be not in NUM has been proposed by Blum and Blum [6]. In [6] reliable EX-learning (in particular, on the set of recursive functions) has been considered. Here, an EX-learner M is said to be reliable iff M, on every recursive function f, either converges to a program for f or diverges (cf. Definition 3). Furthermore, they considered a class $\mathcal{B}$ of approximations to the halting problem K and showed that $\mathcal{B}$ is reliably EX-learnable. This class $\mathcal{B}$ is quite natural and not self-describing. It remained, however, open whether or not $\mathcal{B}$ is in NUM.

After 24 years, our Theorem 2 provides the negative answer. We then aim to investigate further learnability properties of this concrete class. Moreover, we study the dependence of the results obtained for $\mathcal{B}$ for naturally arising variations within the definition of class $\mathcal{B}$.

First, we generalize the approach of Blum and Blum [6] by approximating the characteristic function of any given recursively enumerable set A instead of K (cf. Definition 7). For many non-recursive but recursively enumerable sets A — unfortunately not for all — we could show that the generalized class $\mathcal{U}(A)$ is not in NUM (cf. Theorem 7). Reliable learnability depends on recursion theoretic properties of A: If A is part of a recursively inseparable pair or if A is simple but not hypersimple then $\mathcal{U}(A)$ is reliably EX-learnable (cf. Theorem 10). But if A is hypersimple and not high then $\mathcal{U}(A)$ is not reliably EX-learnable (cf. Theorem 11).

Blum and Blum [6] consider only approximations to the halting problem generated by monotone complexity functions but this restriction seems to be a bit artificial and has not been motivated. We therefore also investigate learnability properties of the classes $\tilde{\mathcal{B}}$ and $\tilde{\mathcal{U}}(A)$ of approximation functions defined by all total complexity functions. In this setting the learnability results depend on particular properties of the complexity measures.

Blum's [5] axioms of an abstract complexity measure turn out to be very flexible and to permit many constructions. There has been a long discussion which additional axioms a natural complexity measure should satisfy (cf., e.g., [15,25,26]). In the context of the present work, we explore new aspects of this old problem. On the one hand, we discovered a "coding condition" defined by requesting $\Phi_i(x) \geq i$ for all $x \in dom(\Phi_i)$ that ensures the EX-learnability of the classes $\tilde{\mathcal{B}}$ as well as of $\tilde{\mathcal{U}}(A)$ for all recursively enumerable sets A (cf. Theorem 12 and Corollary 14).

On the other hand, an aspect of natural complexity measures such as the space used on the work tape of a Turing machine becomes important. That is, now one can modify a complexity function at finitely many places according to a table without changing the rest (we call this the interpolation property of a measure) or that one can do even more involved adaptations (we call measures permitting them regular). For such regular complexity measures the classes $\tilde{\mathcal{B}}$ and $\tilde{\mathcal{U}}(A)$ for all recursively enumerable but non-recursive sets A are not BC-learnable (cf. Theorems 13 and 15).

For giving a formal approach to "natural learnability," Bārzdiņš suggested to look at versions of learning that are closed under computable transformations (cf. [24,34]). For example, a class $\mathcal{U}$ is *robustly EX-learnable*, iff, for every computable operator Θ such that $\Theta(\mathcal{U})$ is a class of recursive functions, the class $\Theta(\mathcal{U})$ is EX-learnable, too (cf. Definition 5). There have been many discussions which operators are admissible in this context (cf., e.g., [12,17,19,24,29,34]). At the end, it turned out to be most suitable to consider only general recursive operators, that is, operators which map every total function to a total one. The resulting notion of robust EX-learning is the most general one among all notions of robust EX-inference.

This notion has been criticized from two sides. On the one hand, the class $\mathcal{C}$ of all total complexity functions Φ_i is not robustly EX-learnable. This can be seen by using Theorem 2.4 and Corollary 2.6 in [29], because there is no recursive function that bounds every function in $\mathcal{C}$ for all but finitely many arguments. On the other hand Jain *et al.* [19] constructed a class $\mathcal{S}$ of functions that pointwise converge to the constant zero function 0^∞. Each function $\varphi_e \in \mathcal{S}$ coincides with 0^∞ on all arguments below e. This is a topological coding of the index e into the function φ_e and no general recursive operator can erase the coded information to an extent which makes EX-learning impossible.

Our analysis of the robust EX-learnability of the class $\mathcal{B}$ and its variants emphasizes this criticism in both directions: For interpolating complexity measures $\mathcal{B}$ is not robustly *EX*-learnable (cf. Theorem 19). On the other hand, for all complexity functions that satisfy the "coding condition" mentioned above, $\mathcal{B}$ and, more generally, even $\tilde{\mathcal{B}}$ as well as every class $\tilde{\mathcal{U}}(A)$ turn out to be robustly *EX*-learnable (cf. Theorem 16 and Corollaries 17 and 18).

2 Preliminaries

Unspecified notations follow Rogers [30]. $\mathbb{N} = \{0, 1, 2, \ldots\}$ and $\mathbb{N}^*$ denote the set of all natural numbers and the set of all finite sequences of natural numbers, respectively. $\{0,1\}^*$ stands for the set of all finite $\{0,1\}$-valued sequences and for all $x \in \mathbb{N}$ we use $\{0,1\}^x$ for the set of all $\{0,1\}$-valued sequences of length x. Let $\emptyset$, $\in$, $\subset$, $\subseteq$, $\supset$, and $\supseteq$, denote the empty set, element of, proper subset, subset, proper superset, and superset, respectively. For any set A we write $\wp(A)$ and $\overline{A}$ for the power set of A and the complement of A, respectively.

The classes of all partial recursive and recursive functions of one, and two arguments over $\mathbb{N}$ are denoted by $\mathcal{P}$, $\mathcal{P}^2$, $\mathcal{R}$, and $\mathcal{R}^2$, respectively. A function $f \in \mathcal{P}$ is said to be monotone provided for all $x, y \in \mathbb{N}$ with $x \leq y$ we have, if both $f(x)$ and $f(y)$ are defined then $f(x) \leq f(y)$. $\mathcal{R}_{0,1}$ and $\mathcal{R}_{mon}$ denotes the set of all $\{0,1\}$-valued recursive functions and of all monotone recursive functions, respectively.

Furthermore, using a fixed encoding $\langle \cdots \rangle$ of $\mathbb{N}^*$ onto $\mathbb{N}$ we write f^n instead of $\langle (f(0), \ldots, f(n)) \rangle$, for any $n \in \mathbb{N}$ and $f \in \mathcal{R}$. Sometimes it will be suitable to identify a recursive function with the sequence of its values, e.g., let $\alpha = (a_0, \ldots, a_k) \in \mathbb{N}^*$, $j \in \mathbb{N}$, and $p \in \mathcal{R}_{0,1}$; then we write $\alpha j p$ to denote the function f for which $f(x) = a_x$, if $x \leq k$, $f(k+1) = j$, and $f(x) = p(x - k - 2)$, if $x \geq k + 2$. Furthermore, let $g \in \mathcal{P}$ and $\alpha \in \mathbb{N}^*$; we write $\alpha \preceq g$ iff α is a prefix of the sequence of values associated with g, i.e., for all $x \leq k$, $g(x)$ is defined and $g(x) = a_x$. If $\mathcal{U} \subseteq \mathcal{R}$, then we denote by $[\mathcal{U}]$ the set of all prefixes of all functions from $\mathcal{U}$.

Any function $\psi \in \mathcal{P}^2$ is called a numbering. Moreover, let $\psi \in \mathcal{P}^2$, then we write ψ_i instead of $\lambda x.\psi(i,x)$ and set $\mathcal{P}_\psi = \{\psi_i \mid i \in \mathbb{N}\}$ as well as $\mathcal{R}_\psi = \mathcal{P}_\psi \cap \mathcal{R}$. Consequently, if $f \in \mathcal{P}_\psi$, then there is a number i such that $f = \psi_i$. If $f \in \mathcal{P}$ and $i \in \mathbb{N}$ are such that $\psi_i = f$, then i is called a ψ–program for f. Let ψ be any numbering, and $i, x \in \mathbb{N}$; if $\psi_i(x)$ is defined (abbr. $\psi_i(x)\downarrow$) then we also say that $\psi_i(x)$ *converges*. Otherwise, $\psi_i(x)$ is said to *diverge* (abbr. $\psi_i(x)\uparrow$).

A numbering $\varphi \in \mathcal{P}^2$ is called a Gödel numbering or acceptable numbering (cf. [30]) iff $\mathcal{P}_\varphi = \mathcal{P}$, and for any numbering $\psi \in \mathcal{P}^2$, there is a $c \in \mathcal{R}$ such that $\psi_i = \varphi_{c(i)}$ for all $i \in \mathbb{N}$. By $G\ddot{o}d$ we denote the set of all Gödel numberings. As usual, we define the halting problem [4] to be the set $K = \{i \mid i \in \mathbb{N},\ \varphi_i(i)\downarrow\}$.

Definition 1 (Blum [5]). A pair (φ, Φ) is called *complexity measure* iff $\varphi \in G\ddot{o}d$ and the *complexity function* $\Phi \in \mathcal{P}^2$ satisfies the following axioms.

(1) $dom(\varphi_i) = dom(\Phi_i)$ for all $i \in \mathbb{N}$;
(2) $\{(i, x, y) \mid \ i, x, y \in \mathbb{N},\ \Phi_i(x) \leq y\}$ is recursive.

Furthermore, for every numbering $\psi \in \mathcal{P}^2$, we say that ψ *has a recursive graph* iff $\{(e, x, t) \mid \psi_e(x)\downarrow \leq t\}$ is recursive.

If $(\hat{\varphi}, \hat{\Phi})$ is a complexity measure and if $\psi \in \mathcal{P}^2$ is a function with recursive graph, then one can construct the *join* $\Phi \in \mathcal{P}^2$ of $\hat{\Phi}$ and ψ by setting $\Phi_{2i} = \hat{\Phi}_i$ and $\Phi_{2i+1} = \psi_i$ for all $i \in \mathbb{N}$. Analogously, we define the join φ of $\hat{\varphi}$ and ψ, i.e., $\varphi_{2i} = \hat{\varphi}_i$ and $\varphi_{2i+1} = \psi_i$ for all $i \in \mathbb{N}$. Note that (φ, Φ) is then again a complexity measure. The following lemma summarizes this fact.

Lemma 1. *For every complexity measure $(\hat{\varphi}, \hat{\Phi})$ and every numbering $\psi \in \mathcal{P}^2$ with recursive graph, there is a further complexity measure (φ, Φ) such that φ and Φ are the join of $\hat{\varphi}$ and ψ and of $\hat{\Phi}$ and ψ, respectively.*

Furthermore, let $NUM = \{\mathcal{U} \mid (\exists \psi \in \mathcal{R}^2)\,[\mathcal{U} \subseteq \mathcal{P}_\psi]\}$ denote the family of all subsets of all recursively enumerable classes of recursive functions.

Now we are ready to define the concepts of learning mentioned in the introduction.

Definition 2. Let $\mathcal{U} \subseteq R$, let $\varphi \in G\ddot{o}d$, and let $M \in \mathcal{P}$ be a machine.

(a) (Gold [13]) M is an *EX-learner* for $\mathcal{U}$ iff, $M(f^n)$ is defined for each function $f \in \mathcal{U}$ and all $n \in \mathbb{N}$, and M converges syntactically to f in the sense that there is a $j \in \mathbb{N}$ with $\varphi_j = f$ and $j = M(f^n)$ for all but finitely many $n \in \mathbb{N}$.

(b) (Angluin [2]) M is a *conservative EX-learner* for $\mathcal{U}$ iff M is an EX-learner for $\mathcal{U}$ and M makes in addition only *necessary hypothesis changes* in the sense such that, for all $\sigma,\ \eta \in \mathbb{N}^*$ with $\sigma, \sigma\eta \in [\mathcal{U}]$, whenever $M(\sigma\eta) \neq M(\sigma)$ then the program $M(\sigma)$ is inconsistent with the data $\sigma\eta$ by either $\varphi_{M(\sigma)}(x)\uparrow$ or $\varphi_{M(\sigma)}(x)\downarrow \neq \sigma\eta(x)$ for some $x \in dom(\sigma\eta)$.

(c) (Bārzdiņš [4], Case and Smith [8]) M is a *BC-learner* for $\mathcal{U}$ iff, $M(f^n)$ is

[4] Note that this definition depends on the Gödel numbering φ, but none of the properties of the halting problem used later does so.

defined for each function $f \in \mathcal{U}$ and all $n \in \mathbb{N}$, and M converges semantically to f in the sense that $\varphi_{M(f^n)} = f$ for all but finitely many $n \in \mathbb{N}$.

A class $\mathcal{U}$ is EX-learnable iff it has a recursive EX-learner and EX denotes the family of all EX-learnable function classes. Similarly we define when a class is conservatively EX-learnable or BC-learnable. We write BC for the family of all BC-learnable function classes. Note that the learning models defined within Definition 2 *do not depend* on the particular choice of the Gödel numbering φ.

Since, by the definition of convergence, only finitely many data of f were seen by the learner up to the (unknown) point of convergence, whenever a learner M succeeds on f, some form of learning must have taken place. For this reason, from here on the terms *infer*, *learn*, and *identify* are used interchangeably.

Note that $EX \subset BC$ (cf. [8]). As far as we are aware of, it has been open whether or not conservative learning constitutes a restriction for EX-learning of recursive functions. Before answering this question, it should be noted that in the above definition conservativeness is only required on initial segments belonging to some function $f \in \mathcal{U}$ of the target class $\mathcal{U}$. This version of conservative learning is usually referred to as *class conservative* EX-learning (cf. [18]). If we require a learner to be conservative on all initial segments $\alpha \in \mathbb{N}^*$ then we obtain *global conservative* EX-learning. Now, we can prove the following.

PROPOSITION 1. *For every class $\mathcal{U} \subseteq \mathcal{R}$ the following is equivalent.*
(a) $\mathcal{U}$ *is EX-learnable.*
(b) $\mathcal{U}$ *is class conservatively EX-learnable.*
(c) $\mathcal{U}$ *is globally conservatively EX-learnable.*

Proof. Clearly, (c) implies (b) and (b) implies (a). Thus, it suffices to show that every EX-learnable class is also globally conservatively EX-inferable. For seeing this, we recall the following characterization. A class $\mathcal{U}$ is EX-learnable iff there exist a numbering $\psi \in \mathcal{P}^2$ and a function $u \in \mathcal{R}^2$ such that $\mathcal{U} \subseteq \mathcal{P}_\psi$ and for all $i, j \in \mathbb{N}$, $i \neq j$ implies $\psi_i \neq_{u(i,j)} \psi_j$ (cf. Wiehagen [31]). Here $\psi_i =_n \psi_j$, $n \in \mathbb{N}$, means that for all $x \leq n$ either both $\psi_i(x)$ and $\psi_j(x)$ are defined and equal or both $\psi_i(x)\uparrow$ and $\psi_j(x)\uparrow$.

Now, let $\mathcal{U} \in EX$, let (φ, Φ) be a complexity measure, let ψ and u as above, and let $c \in \mathcal{R}_{mon}$ be such that $\psi_i = \varphi_{c(i)}$ for all $i \in \mathbb{N}$. On every input $f^n \in [\mathcal{R}]$ the desired learner M behaves as follows. If $n = 0$ then output $c(0)$. Otherwise, let $c(i) = M(f^{n-1})$. For all $j \leq n$ such that $i \neq j$ and $u(i,j) \leq n$ check whether or not $\Phi_{c(i)}(y) \leq n$ for all $y \leq u(i,j)$. If j passes this test successfully then check whether or not $\psi_j =_{u(i,j)} f$. If and only if a j has been found that passes both tests then output $c(i+1)$.

Clearly, $M \in \mathcal{R}$ and if M changes its hypothesis from $c(i)$ to $c(i+1)$ then, by the choice of u, the condition $\psi_j =_{u(i,j)} f$ implies that i is definitely wrong and, moreover, $\psi_i \neq_{u(i,j)} \psi_j =_{u(i,j)} f$ guarantees that either $\psi_i(y)\uparrow$ or $\psi_i(y)\downarrow \neq f(y)$ for some $y \leq u(i,j) \leq n$. Hence, M is globally conservative.

Now, let $f \in \mathcal{U}$ and let m be the least ψ-program for f. Then, for $i = m$, no $j > m$ can pass the test above. On the other hand, as long as M outputs some $c(i)$ with $i < m$, there is at least one j (namely m) that will pass the test for n sufficiently large. Thus, M changes its hypothesis from $c(i)$ to $c(i+1)$. Hence M EX-infers $\mathcal{U}$. ∎

Next, we define reliable inference. Intuitively, a learner M is reliable provided it converges if and only if it learns. There are several variants of reliable learning, so we will give a justification of our choice below.

DEFINITION 3 (Blum and Blum [6], Minicozzi [27]). Let $\mathcal{U} \subseteq R$ and let $\varphi \in$ *Göd*; then $\mathcal{U}$ is said to be *reliably EX-learnable* if there is a machine $M \in \mathcal{R}$ such that

(1) M EX-learns $\mathcal{U}$ and
(2) for all $f \in \mathcal{R}$, if the sequence $(M(f^n))_{n\in\mathbb{N}}$ converges, say to j, then $\varphi_j = f$.

By $\mathcal{R}EX$ we denote the family of all reliably EX-learnable function classes.

Note that one can replace the condition "$f \in \mathcal{R}$" in (2) of Definition 3 by "$f \in \mathcal{P}$" or "all total f." This would result in a different model of reliable learning, say $\mathcal{P}EX$ and $\mathcal{T}EX$, respectively. In particular, then for every $\mathcal{U} \subseteq \mathcal{R}_{0,1}$ such that $\mathcal{U} \in \mathcal{P}EX$ or $\mathcal{U} \in \mathcal{T}EX$ one has $\mathcal{U} \in NUM$ (cf. [6,14,32]). On the other hand, there are classes $\mathcal{U} \subseteq \mathcal{R}_{0,1}$ such that $\mathcal{U} \in \mathcal{R}EX \setminus NUM$ (cf. [14]). As a matter of fact, our Theorem 2 below together with Blum and Blum's [6] result $\mathcal{B} \in \mathcal{R}EX$ provides a much easier proof of the same result than Grabowski [14].

Finally, we define robust-EX-learning. This definition involves the notion of general recursive operators. Intuitively, a general recursive operator is a computable mapping that maps functions over $\mathbb{N}$ to functions over $\mathbb{N}$. Furthermore, it is required that every total function is mapped to a total function. Formally, a general recursive operator is defined as follows. Let $\vartheta_0, \vartheta_1, \ldots$ be the canonical enumeration of all finite functions.

DEFINITION 4 (Rogers [30]). A mapping Θ from partial functions to partial functions is called a *partial recursive operator* iff there is a recursively enumerable set $W \subset \mathbb{N}^3$ such that for any $x, y \in \mathbb{N}$ it holds that $\Theta(f)(x) = y$ iff there is z with $(x, y, z) \in W$ and f extends the finite function ϑ_z.

Furthermore, Θ is called a *general recursive operator* iff for every $x \in \mathbb{N}$ and every total function f there are unique y and z such that $(x, y, z) \in W$ and f extends ϑ_z, that is, a partial recursive operator Θ is general recursive iff Θ maps every total function to a total function.

For more information about general recursive operators the reader is referred to [16,28,33].

Definition 5 (Jain, Smith and Wiehagen [19]). Let $\mathcal{U} \subseteq \mathcal{R}$; then $\mathcal{U}$ is said to be *robustly EX-learnable* if $\Theta(\mathcal{U})$ is EX-learnable for every general recursive operator Θ. By robust-EX we denote the family of all robustly EX-learnable function classes.

3 Approximating the Halting Problem

Within this section, we deal with Blum and Blum's [6] original class $\mathcal{B}$ mentioned in the introduction. First, we define the class of approximations to the halting problem that has been considered in [6].

Definition 6. Let (φ, Φ) be any complexity measure, and let $\tau \in \mathcal{R}$ be such that for all $i \in \mathbb{N}$

$$\varphi_{\tau(i)}(x) = \begin{cases} 1, & \text{if } \Phi_i(x)\downarrow \text{ and } \Phi_x(x) \leq \Phi_i(x), \\ 0, & \text{if } \Phi_i(x)\downarrow \text{ and } \neg[\Phi_x(x) \leq \Phi_i(x)], \\ \uparrow, & \text{otherwise.} \end{cases}$$

Now, we set $\mathcal{B} = \{\varphi_{\tau(i)} \mid i \in \mathbb{N} \text{ and } \Phi_i \in \mathcal{R}_{mon}\}$.

Note that the class $\mathcal{B}$ depends on the complexity measure (φ, Φ). However, many of our results do *not depend* on the particular choice of (φ, Φ). Therefore we usually do not write $\mathcal{B}^{(\varphi,\Phi)}$ instead of $\mathcal{B}$ except in case the result we want to prove *does* depend on particular properties of (φ, Φ).

Blum and Blum [6] have shown $\mathcal{B} \in \mathcal{R}EX$ but left it open whether or not $\mathcal{B} \in NUM$. It is *not*, as our next theorem shows.

Theorem 2. $\mathcal{B} \notin NUM$.

Proof. First, recall that K is part of a recursively inseparable pair (cf. [28, Exercise III.6.23.(a)]). That is, there is an r.e. set H such that $K \cap H = \emptyset$ and no recursive set separates H from K. So given any recursive set $A \supseteq H$, this set cannot be disjoint to K and we even have $|A \cap K| = \infty$. Now, we fix any enumeration $k_0, k_1, k_2, \ldots$ and $h_0, h_1, h_2, \ldots$ of K and H, respectively. Suppose to the contrary, that there exists a numbering $\psi \in \mathcal{R}^2$ such that

$\mathcal{B} \subseteq \mathcal{R}_\psi$. Next, we define for each ψ_e a function $g_e \in \mathcal{P}$ as follows. For all $e, x \in \mathbb{N}$ let

$g_e(x) =$ "Search for the least n such that for $n = s + y$ either (A), (B) or (C) happens:
(A) $y = h_s \wedge \psi_e(y) = 1$,
(B) $y = k_s \wedge \psi_e(y) = 0 \wedge y > x$,
(C) $\psi_e(y) > 1$.
If (A) happens first, then let $g_e(x) = s + y$.
If (B) happens first, then let $g_e(x) = \Phi_y(y) + y$.
If (C) happens first, then let $g_e(x) = 0$."

Claim 1. *$g_e \in \mathcal{R}$ for all $e \in \mathbb{N}$.*

Clearly, if there is at least one y such that $\psi_e(y) > 1$, then $g_e \in \mathcal{R}$. Now assume $\psi_e \in \mathcal{R}_{0,1}$ and suppose that there is an $x \in \mathbb{N}$ with $g_e(x)\uparrow$. Then there are no s, y such that $y = h_s$ and $\psi_e(y) = 1$. Consequently, the set $L = \{y|\ y \in \mathbb{N} \wedge \psi_e(y) = 0\}$ is recursive and $L \supseteq H$. Thus, $|L \cap K| = \infty$, and therefore there must be a $y > x$ such that $\psi_e(y) = 0$ and an $s \in \mathbb{N}$ with $y = k_s$. Consequently (B) must happen, and since $y = k_s$, we conclude $\Phi_y(y)\downarrow$. Hence, $g_e(x)\downarrow$, too, a contradiction. This proves Claim 1.

Claim 2. *Let e be any number such that $\psi_e = \varphi_{\tau(i)}$ for some $\varphi_{\tau(i)} \in \mathcal{B}$. Then $g_e(x) > \Phi_i(x)$ for all $x \in \mathbb{N}$.*

Assume any i, e as above, and consider the definition of $g_e(x)$. Suppose $g_e(x) = s + y$ for some s, y such that $y = h_s$ and $\psi_e(y) = 1$. Since $\psi_e(y) = \varphi_{\tau(i)}(y) = 1$ implies $\Phi_y(y) \leq \Phi_i(y)$, and hence $y \in K$, we get a contradiction to $K \cap H = \emptyset$. Thus, this case cannot happen.

Consequently, in the definition of $g_e(x)$ condition (B) must have happened. Thus, some s, y such that $y > x$, $y = k_s$ and $\psi_e(y) = 0$ have been found. Since $y = k_s$, we again conclude $\Phi_y(y)\downarrow$ and thus $g_e(x) > \Phi_y(y)$. Furthermore, because of $\psi_e(y) = \varphi_{\tau(i)}(y) = 0$, we directly obtain that $\Phi_i(y) < \Phi_y(y)$ by the definition of $\varphi_{\tau(i)}$. Now, putting it all together, we get $g_e(x) > \Phi_y(y) > \Phi_i(y) \geq \Phi_i(x)$, since $y > x$ and $\Phi_i \in \mathcal{R}_{mon}$. This proves Claim 2.

Claim 3. *For every $b \in \mathcal{R}$ there exists an $i \in \mathbb{N}$ such that $\Phi_i \in \mathcal{R}_{mon}$ and $b(x) < \Phi_i(x)$ for all $x \in \mathbb{N}$.*

Let $r \in \mathcal{R}$ be such that for all $j, x \in \mathbb{N}$ we have

$$\varphi_{r(j)} = \begin{cases} 0, & \text{if } \Phi_j(y) \text{ is defined for all } y < x \\ & \text{and either } \Phi_j(x) \text{ is undefined or } \Phi_j(x) \\ & \text{satisfies } \Phi_j(x) > \max(\{b(x)\} \cup \{\Phi_j(y)|\ \ y < x\}); \\ \uparrow, & \text{otherwise.} \end{cases}$$

Now take i such that $\varphi_i = \varphi_{r(i)}$ according to the fixed point theorem [30]. One sees that, on the one hand, whenever $\varphi_i(y)$ is defined for all $y < x$, then $\varphi_i(x)$ is undefined if and only if $\Phi_i(x)$ is defined and either below $b(x)$ or below $\Phi_i(y)$ for some $y < x$. But, on the other hand, by the definition of the measure Φ_i, it holds that $\Phi_i(x)$ is defined iff $\varphi_i(x)$ is defined. So it follows that φ_i is total and thus Φ_i is a monotone majorant of b. This proves Claim 3.

Finally, by Claim 1, all $g_e \in \mathcal{R}$, and thus there is a function $b \in \mathcal{R}$ such that $b(x) \geq g_e(x)$ for all $e \in \mathbb{N}$ and all but finitely many $x \in \mathbb{N}$ (cf. [6]). Together with Claim 2, this function b would contradict Claim 3, and hence $\mathcal{B} \notin NUM$. ∎

Next, we ask in which sense the class $\mathcal{B}$ is learnable. On the one hand, we show $\mathcal{B}$ to be $\mathcal{R}EX$-learnable. It should be noted that $\mathcal{B} \in \mathcal{R}EX$ has already been proved by Blum and Blum [6]. However, their proof is done by showing that there exists a general recursive operator Θ such that every function in $\mathcal{B}$ is everywhere Θ-compressed. Then the result follows by the a posteriori characterization of $\mathcal{R}EX$-learning (cf. [6], Theorem 4). In contrast, we provide a direct proof for $\mathcal{B} \in \mathcal{R}EX$. On the other hand, we also prove directly that $\mathcal{B}$ is conservatively EX-inferable.

THEOREM 3. *$\mathcal{B}$ is $\mathcal{R}EX$-learnable and conservatively EX-inferable.*

Proof. First, we show $\mathcal{B} \in \mathcal{R}EX$. Note that for every $f \in \mathcal{R}$ and $n \in \mathbb{N}$ one can effectively find a φ-program i such that $\varphi_i = f^n 0^\infty$. Moreover, we require this procedure to do the following for all $f, g \in \mathcal{R}$ and $m, n \in \mathbb{N}$: if $g^m 0^\infty = f^n 0^\infty$ as well as $\varphi_i = f^n 0^\infty$ and $\varphi_j = g^m 0^\infty$ then the indices given by our effective procedure should be equal: $i = j$. We refer to the such found i as to the canonical index of $f^n 0^\infty$. Furthermore, let the set H be as in the proof of Theorem 2, and fix again any enumeration $k_0, k_1, k_2, \ldots$ of K. We use K_n to denote $\{k_0, \ldots, k_{n-1}\}$, i.e., the set of the first n enumerated elements of K.

The desired machine M is defined as follows. Let $f \in \mathcal{R}$ and $n \in \mathbb{N}$; then

$M(f^n) =$ "Spend half of the time doing (A) and half doing (B).

(A) Check whether there exists an $x \leq n$ such that $f(x) > 1$ or $f(x) = 1$ and $x \in H$. If and when such an x is found, output the canonical index of $f^n 0^\infty$.

(B) Compute K_n and determine the largest $b \leq n$ with $f(b) = 0$ and $b \in K_n$. If there is no such b then output the canonical index of $f^n 0^\infty$.

Otherwise, search the least $i \leq n$ such that $\Phi_i(b)\downarrow \leq \Phi_b(b)$, $\Phi_i(x) \leq \Phi_i(b)$ and $\varphi_{\tau(i)}(x) = f(x)$ for all $x \leq b$. If and when such an i is found, output $\tau(i)$. If the test has been failed for all $i \leq n$, output the canonical index of $f^n 0^\infty$."

It remains to show that M behaves correctly.

Claim 1. $M \in \mathcal{R}$.

It suffices to show that $M(f^n)$ is defined for all $n \in \mathbb{N}$ and $f \in \mathcal{R}$. Clearly, if there is no $b \leq n$ with $f(b) = 0$ and $b \in K_n$ then $M(f^n)$ is already defined. Otherwise, $b \in K_n$ and therefore $\Phi_b(b)\downarrow$. Hence, one can effectively check whether or not there is an $i \leq n$ with $\Phi_i(b)\downarrow \leq \Phi_b(b)$ and $\Phi_i(x)\downarrow \leq \Phi_i(b)$ for all $x \leq b$. For every i passing this test we may conclude that $\varphi_{\tau(i)}(x)\downarrow$ for all $x \leq b$ by the definition of $\varphi_{\tau(i)}$. Thus, the condition $\varphi_{\tau(i)}(x) = f(x)$ for all $x \leq b$ can be effectively tested, too. Consequently, $M(f^n)$ is again defined. This proves Claim 1.

Claim 2. *M is reliable.*

Suppose any $f \in \mathcal{R}$ such that $(M(f^n))_{n\in\mathbb{N}}$ converges, say to j. We have to show that $\varphi_j = f$. First, if M converges by outputting only canonical indices of $f^n 0^\infty$ beyond some point, then f must be of the form $\alpha 0^\infty$ for some $\alpha \in \mathbb{N}^*$. Hence, in this case we obviously have $\varphi_j = f$.

Otherwise, M has to converge to some $\tau(i)$. If there are infinitely many $b \in K$ such that $f(b) = 0$, then M indeed verifies in the limit that $f_{\tau(i)} = f$, and we are again done. Finally, suppose that there are only finitely many $b \in K$ with $f(b) = 0$. Since M converges to some $\tau(i)$, it must verify in the limit for all $x \in H$ that $f(x) = 0$. To see this, suppose that there is at least one $x \in H$ with $f(x) = 1$. Then, finding this x would take a finite number of steps, say n_0. Consequently, for $n > n_0$, the computation of $M(f^n)$ would finish in (A), since enumerating K_n takes at least $n > n_0$ steps. Thus, the set $S = \{f(x) |\ f(x) = 0\}$ is recursive and $S \supseteq H$. By the choice of H, we get $|K \cap S| = \infty$, a contradiction. This proves Claim 2.

Claim 3. *M learns $\mathcal{B}$.*

Let $f \in \mathcal{B}$; then for every $x \notin K$ we have $f(x) = 0$ by the definition of $\mathcal{B}$. Moreover, $\mathcal{B} \subseteq \mathcal{R}_{0,1}$, and hence for every $f \in \mathcal{B}$ the computation in (A) can never terminate. Finally, we again have $S = \{f(x) |\ f(x) = 0\} \supseteq H$ and S is recursive. Thus, for every $f \in \mathcal{B}$ there are infinitely many $b \in K$ such that $f(b) = 0$. Since $f \in \mathcal{B}$, there must be a j such that $\varphi_{\tau(j)} = f$ and $\Phi_j \in \mathcal{R}_{mon}$. Since $f(b) = 0$, we conclude $\Phi_j(b)\downarrow \leq \Phi_b(b)$, and because of $\Phi_j \in \mathcal{R}_{mon}$ we also have $\Phi_j(x) \leq \Phi_j(b)$ for all $x \leq b$. Thus, M converges to some $\tau(i)$ with $i \leq j$. Since M is reliable, it learns f. This proves Claim 3.

Now, putting it all together, we have $\mathcal{B} \in \mathcal{R}EX$.

Since $\mathcal{R}EX \subset EX$, Proposition 1 immediately implies that $\mathcal{B}$ is conservatively EX-learnable, too. Nevertheless, we also give a direct proof which may be interesting in its own right. For doing this, some new ideas are needed. We cannot use the hypothesis space $(\varphi_{\tau(i)})_{i\in\mathbb{N}}$ and the consistency check in (B) above, since this may result in a non-conservative hypothesis change. For example, it is well conceivable that there are i, j with $i \neq j$ and $\varphi_{\tau(i)} = \varphi_{\tau(j)}$ but $\Phi_i \notin \mathcal{R}_{mon}$ while $\Phi_j \in \mathcal{R}_{mon}$. Since we cannot check consistency without knowing that all values must be defined (remember that $\mathcal{B} \notin NUM$), the learner may be forced to change its hypothesis from i to j. Therefore, we first define a new hypothesis space $(\varphi_{\kappa(i)})_{i\in\mathbb{N}}$ such that for every $\Phi_i \notin \mathcal{R}_{mon}$ the condition $\varphi_{\kappa(i)} \notin \mathcal{B}$ is fulfilled by construction. Let $\kappa \in \mathcal{R}$ be such that for all $i \in \mathbb{N}$

$$\varphi_{\kappa(i)}(x) = \begin{cases} 1, & \text{if } \Phi_i(x)\downarrow \ \wedge \Phi_x(x) \le \Phi_i(x) \wedge \\ & (\forall y < x)\,[\Phi_i(y) \le \Phi_i(y+1)], \\ 0, & \text{if } \Phi_i(x)\downarrow \ \wedge \neg[\Phi_x(x) \le \Phi_i(x)] \wedge \\ & (\forall y < x)\,[\Phi_i(y) \le \Phi_i(y+1)], \\ \uparrow, & \text{otherwise.} \end{cases}$$

Note that $\mathcal{B} = \{\varphi_{\kappa(i)} \mid i \in \mathbb{N} \text{ and } \Phi_i \in \mathcal{R}_{mon}\}$.

Next, we define the desired learner M. Let $f \in \mathcal{B}$ and let $n \in \mathbb{N}$.

$M(f^n) =$ "Compute K_n and determine the largest $b \le n$ with $f(b) = 0$ and $b \in K_n$. If there is no such b then output the canonical index of $f^n 0^\infty$.

Otherwise, search the least i such that $\Phi_i(b)\downarrow \le \Phi_b(b)$, $\Phi_i(y) \le \Phi_i(y+1)$ for all $y < b$ and $\varphi_{\kappa(i)}(x) = f(x)$ for all $x \le b$. If and when such an i is found, output $\kappa(i)$."

Claim 4. *M conservatively EX-learns $\mathcal{B}$.*

As shown in Claim 3 above, there must be infinitely many b with $f(b) = 0$ and $b \in K_n$. As soon as M has found for the first time such a b, say on input f^n, it will find such a b for every extension f^m of f^n, i.e., for all $m \ge n$. Thus M outputs at most finitely many canonical indices for functions of the form $f^k 0^\infty$. Moreover, the search for the least i must always terminate, since $f \in \mathcal{B}$. Thus, $M(f^n)$ is defined for all $f \in \mathcal{B}$ and $n \in \mathbb{N}$.

Furthermore, M converges to some $\kappa(i)$ with $i \le j$, where j is the least index such that $\varphi_{\kappa(j)} = f$ and $\Phi_j \in \mathcal{R}_{mon}$. Since there are infinitely many b with $f(b) = 0$ and $b \in K_n$, M verifies in the limit $\varphi_{\kappa(i)} = f$, and thus EX-learns f.

It remains to show that M is conservative. As long as M outputs only canonical indices, we are done. Now, suppose any n and i, j such that

$$\kappa(i) = M(f^n) \neq M(f^{n+1}) = \kappa(j) \ .$$

Then M must have found a new $\hat{b}$ with $f(\hat{b}) = 0$ and $\hat{b} \in K_{n+1}$ on input f^{n+1}. If $\Phi_i(\hat{b}) \leq \Phi_{\hat{b}}(\hat{b})$ and $\Phi_i(y) \leq \Phi_i(y+1)$ for all $y < \hat{b}$, then we are done, since in this case M must have found some $x \leq \hat{b}$ such that $\varphi_{\kappa(i)}(x)\downarrow \neq f(x)$.

Next, suppose $\Phi_i(\hat{b}) \leq \Phi_{\hat{b}}(\hat{b})$ but $\neg[\Phi_i(y) \leq \Phi_i(y+1)]$ for some $y < \hat{b}$. By the construction of $\varphi_{\kappa(i)}$ we may now conclude that $\varphi_{\kappa(i)}(\hat{b})\uparrow$, and thus we are done.

Finally, assume $\neg[\Phi_i(\hat{b}) \leq \Phi_{\hat{b}}(\hat{b})]$. If $\Phi_i(\hat{b})\downarrow$ then $\varphi_{\kappa(i)}(\hat{b}) = 1$ by the definition of $\mathcal{B}$, a contradiction. Hence $\Phi_i(\hat{b})$ diverges and so does $\varphi_{\kappa(i)}(\hat{b})$. Therefore the hypothesis change has been done conservatively. ∎

Theorems 2 and 3 immediately allow the following separation, thus reproving Grabowski's [14] Theorem 5.

COROLLARY 4. $NUM \cap \wp(\mathcal{R}_{0,1}) \ \subset \ \mathcal{R}EX \cap \wp(\mathcal{R}_{0,1})$

Looking at Theorem 3, the following question arises naturally. Does there exist a learner identifying $\mathcal{B}$ *simultaneously* reliably and conservatively? As a matter of fact, the two general learning machines established in Blum and Blum [6] within the proofs of their characterization theorems for $\mathcal{R}EX$ do *not* have this property. The affirmative answer is provided by our next proposition.

PROPOSITION 2. *Let $U \subseteq \mathcal{R}$ be any class such that $U \in \mathcal{R}EX$. Then there exists a learner which simultaneously infers U reliably and is globally conservative.*

Proof. Since $U \in \mathcal{R}EX$, there is a machine $\widehat{M}$ reliably inferring U. The desired learner M uses $\widehat{M}$ as subroutine. The main problem one has to overcome is to make sure that M behaves also conservatively on functions f that are not learned by $\widehat{M}$. For that purpose, we define a new hypothesis space as follows. Let $\ell \in \mathcal{R}$ be one-one such that for all $f \in \mathcal{R}$ and all $k, n, x \in \mathbb{N}$

$$\varphi_{\ell(f^n,k)}(x) = \begin{cases} f(x), & \text{if } x \leq n \\ \varphi_k(x), & \text{if } x > n \ \wedge \ \varphi_k(y)\downarrow \ \text{ for all } y \leq x \\ & \qquad \wedge\, \varphi_k(y) = f(y) \text{ for all } y \leq n \ \wedge \ \widehat{M}(\varphi_k^x) = k, \\ \uparrow, & \text{otherwise.} \end{cases}$$

Now we are ready to define the desired machine M. Let $f \in \mathcal{R}$ and $n \in \mathbb{N}$; then

$$M(f^n) = \text{“Compute } k := \widehat{M}(f^n) \text{ and find the smallest } m \leq n \text{ such that } \widehat{M}(f^m) = \widehat{M}(f^{m+1}) = \cdots = \widehat{M}(f^n)\text{. Output } \ell(f^m, k)\text{.”}$$

Since $\widehat{M} \in \mathcal{R}$, we directly obtain, by construction, $M \in \mathcal{R}$ and that M makes a mind change iff $\widehat{M}$ does.

Claim 1. *M is reliable.*

Suppose any $f \in \mathcal{R}$ such that $(M(f^n))_{n\in\mathbb{N}}$ converges, say to j. Then there exists $m, k \in \mathbb{N}$ such that $j = \ell(f^m, k)$. By construction, we now know that $\widehat{M}(f^m) = k$, and moreover, that $\widehat{M}$ converges on f to k. Since $\widehat{M}$ is reliable, we may conclude $\varphi_k = f$. Consequently, $\varphi_k(x)\downarrow$ for all $x \in \mathbb{N}$ and thus $\varphi_{\ell(f^m,k)} = f$, too. This proves Claim 1.

Claim 2. *M learns U.*

Let $f \in U$; then $\widehat{M}$ identifies f. Consequently, the sequence $(\widehat{M}(f^n))_{n\in\mathbb{N}}$ converges, say to k. Now, consider the smallest number $m \in \mathbb{N}$ such that $\widehat{M}(f^{m+z}) = k$ for all $z \geq 0$. Thus, M converges on f to $\ell(f^m, k)$. Again, since $\varphi_k = f$, we may conclude that $\varphi_{\ell(f^m,k)} = f$, and therefore M learns f. This proves Claim 2.

Claim 3. *M is conservative.*

Assume any $f \in \mathcal{R}$ and any numbers n, $r \in \mathbb{N}$ such that $M(f^n) \neq M(f^{n+r})$. Consequently, there are numbers k, k', $m, m' \in \mathbb{N}$ such that $M(f^n) = \ell(f^m, k)$ and $M(f^{n+r}) = \ell(f^{m'}, k')$. By construction, we may conclude that $\widehat{M}(f^n) = k = \widehat{M}(f^m)$ and that $\widehat{M}(f^{n+r}) = \widehat{M}(f^{m'}) = k' \neq k$. Now, we distinguish the following cases.

Case 1. $\varphi_k = f$.

In this case, $\widehat{M}(\varphi_k^n) = k \neq \widehat{M}(\varphi_k^{n+r})$. Therefore, our construction directly yields $\varphi_{\ell(f^m,k)}(n+r)\uparrow$, and thus M is conservative.

Case 2. $\varphi_k \neq f$.

Now, if $\varphi_k(y)\uparrow$ for some $y \leq n+r$ we are done, since this directly implies $\varphi_{\ell(f^m,k)}(y)\uparrow$.

Next suppose that $\varphi_k(y)\downarrow$ for all $y \leq n+r$. If $\varphi_k(y) = f(y)$ for all $y \leq n+r$ then we conclude that M is conservative as in Case 1. Finally, if $\varphi_k(y) \neq f(y)$ for some $y \leq n+r$ then, by construction, either $\widehat{M}(\varphi_k^m) = \cdots = \widehat{M}(\varphi_k^{n+r})$, and hence $\varphi_{\ell(f^m,k)}(y)\downarrow \neq f(y)$ for some $y \leq n+r$, or $\widehat{M}(\varphi_k^m) \neq \widehat{M}(\varphi_k^{m+z})$

for some $z > 0$, and hence $\varphi_{\ell(f^m,k)}(m+z)\uparrow$. In both case we can conclude that M is conservative. ∎

We postpone studying further learnability properties of the class $\mathcal{B}$ to Sections 6 and 7. Our next goal is to generalize the approach undertaken so far by defining classes $\mathcal{U}(A)$ of approximations to any recursively enumerable (abbr. r.e.) set A and by looking at their learnability. This is done in the following section.

4 Approximating Arbitrary r.e. Sets

The first difficulty we have to overcome is to find a way to define classes of approximations for arbitrary r.e. sets A. The definition of Blum and Blum's [6] class uses $\Phi_x(x)$ in order to measure implicitly the speed by which K is enumerated. Since such a convenient way is not available in general, we have to go back to an arbitrarily fixed recursive enumeration of the set A under consideration. Moreover, Blum and Blum's [6] definition involves implicitly a one-one enumeration of K and therefore this will be also the only condition we impose on any recursive enumeration of the set A.

DEFINITION 7. Let A be any r.e. set, and let $a_0, a_1, \dots$ be any fixed one-one recursive enumeration of A. For every $\Phi_e \in \mathcal{R}$ let

$$f_e(x) = \begin{cases} 1, & \text{if } (\exists s \le \Phi_e(x))\,[a_s = x]; \\ 0, & \text{if } (\forall s \le \Phi_e(x))\,[a_s \ne x]. \end{cases}$$

Now $\mathcal{U}(A)$ consists of all those f_e where $\Phi_e \in \mathcal{R}_{mon}$.

Note that $\mathcal{U}(A)$ also depends on the complexity measure (φ, Φ). We therefore use the same convention as in Section 3, i.e., we do not write $\mathcal{U}(A)^{(\varphi,\Phi)}$ as long as our assertions do not depend on the particular choice of (φ, Φ).

Comparing $\mathcal{U}(K)$ to the original class $\mathcal{B}$ of Blum and Blum [6] one can easily prove the following. For every $f \in \mathcal{B}$ there is a function $g \in \mathcal{U}(K)$ such that for all $x \in \mathbb{N}$ we have $f(x) = 1$ implies $g(x) = 1$. Hence, the approximation g is at least as good as f. The converse is also true, i.e., for each $g \in \mathcal{U}(K)$ there is an $f \in \mathcal{B}$ such that $g(x) = 1$ implies $f(x) = 1$ for all $x \in \mathbb{N}$. Therefore, we consider our new classes of approximations as natural generalizations of Blum and Blum's [6] original definition.

Note that there is a function gen_A which computes for every e a program

$gen_A(e)$ for the function f associated with Φ_e:

$$\varphi_{gen_A(e)}(x) = \begin{cases} 1, & \text{if } \Phi_e(x)\!\downarrow \wedge\ (\exists s \le \Phi_e(x))\,[a_s = x] \\ & \qquad \wedge\ (\forall y < x)\,[\Phi_e(y) \le \Phi_e(y+1)]; \\ 0, & \text{if } \Phi_e(x)\!\downarrow \wedge\ (\forall s \le \Phi_e(x))\,[a_s \ne x] \\ & \qquad \wedge\ (\forall y < x)\,[\Phi_e(y) \le \Phi_e(y+1)]; \\ \uparrow, & \text{otherwise, in particular if } \Phi_e(x)\!\uparrow. \end{cases}$$

Next, we deal with the learnability properties of these classes $\mathcal{U}(A)$. If A is recursive, everything is clear, since we have the following result.

THEOREM 5. *If A is recursive then $\mathcal{U}(A) \in NUM$.*

Proof. Let A be recursive, and let $a_0, a_1, \ldots$ be any one-one enumeration of A. Then there is a recursive function g such that

$$g(x) = \begin{cases} s, & \text{if } x \in A \text{ and } x = a_s; \\ \infty, & \text{otherwise } (x \notin A). \end{cases}$$

Now one can construct an enumeration $\psi_0, \psi_1, \ldots$ such that

$$\psi_e(x) = \begin{cases} 0, & \text{if } g(x) = \infty \text{ or } \Phi_e(x)\!\downarrow < g(x); \\ 1, & \text{if } g(x) < \infty \text{ and } \neg[\Phi_e(x)\!\downarrow < g(x)]. \end{cases}$$

By “lazy evaluation” we have that $\psi_e(x) = 0$ whenever $g(x) = \infty$. So every ψ_e is total. Furthermore, the ψ_e are uniformly recursive since, by the definition of a complexity measure, the set $\{(x, y, e) \mid x, y, e \in \mathbb{N},\ \Phi_e(x) \le y\}$ is recursive. If f_e (as defined in Definition 7) exists then Φ_e is total and “$\neg[\Phi_e(x)\!\downarrow < g(x)]$” is equivalent to “$g(x) \le \Phi_e(x)$”. It follows that $f_e = \psi_e$. Thus, $\mathcal{U}(A) \subseteq \{\psi_0, \psi_1, \ldots\}$ and $\mathcal{U}(A) \in NUM$. ∎

The direct generalization of Theorem 2 would be that $\mathcal{U}(A)$ is not in NUM for every non-recursive r.e. set A. Unfortunately, there are some special cases where this is still unknown to us.

Nevertheless we obtained many intermediate results which support the conjecture that $\mathcal{U}(A)$ is not in NUM for any non-recursive r.e. set A. In particular, we can show that $\mathcal{U}(A) \notin NUM$ for every r.e. set A that is part of a recursively inseparable pair, that is simple but not hypersimple or that is neither recursive nor high.

Recall that A and B form a recursively inseparable pair if A and B are disjoint but there is no recursive set R with $A \subseteq R$ and $B \subseteq \overline{R}$. In particular, every recursive set R which contains all but finitely many elements of A, intersects with B. A set A is *simple* iff A is both r.e. and infinite, $\overline{A}$ is infinite but there is no infinite recursive set R disjoint to A.

A set A is *hypersimple* iff A is both r.e. and infinite, and there is no function $f \in \mathcal{R}$ such that $f(n) \geq \overline{a}_n$ for all $n \in \mathbb{N}$, where $\overline{a}_0, \overline{a}_1, \ldots$ is the enumeration of $\overline{A}$ in strictly increasing order (cf. Rogers [30]). Using this definition of hypersimple sets, one can easily show the following lemma.

LEMMA 6. *A set* $A \subseteq \mathbb{N}$ *is hypersimple iff*
(a) A *is r.e. and both* A *and* $\overline{A}$ *are infinite*
(b) *for all functions* $g \in \mathcal{R}$ *with* $g(x) \geq x$ *for all* $x \in \mathbb{N}$ *there exist infinitely many* $x \in \mathbb{N}$ *such that* $\{x, x+1, \ldots, g(x)\} \subseteq A$.

Now, we are ready to state the announced theorem.

THEOREM 7. $\mathcal{U}(A)$ *is not in NUM for the following r.e. sets* A.
(a) A *is part of a recursively inseparable pair.*
(b) A *is simple but not hypersimple.*
(c) A *is neither recursive nor high.*

Proof. For the whole proof, let $a_0, a_1, \ldots$ be the fixed one-one recursive enumeration of A. Note that the basic idea for proving (a) and (b) resembles the demonstration of Theorem 2, while for showing (c) a new idea is needed. Now assume by way of contradiction $\mathcal{U}(A) \in NUM$. Thus, there exists a $\psi \in \mathcal{R}^2$ such that $\mathcal{U}(A) \subseteq \mathcal{R}_\psi$.

First, we prove (a). Let B be a recursively enumerable set such that A and B form a recursively inseparable pair and let $b_0, b_1, \ldots$ be any fixed recursive enumeration of B. For each ψ_e we define a function $F_e \in \mathcal{P}$ as follows.

$F_e(x) =$ "Search for the least s such that either (A) or (B) happens:
(A) $\psi_e(b_s) = 1$ or $\psi_e(a_s) = 0$ and $a_s \geq x$,
(B) $\psi_e(s) > 1$.
If (A) happens first, then set $F_e(x) = s$.
If (B) happens first, then let $F_e(x) = 0$."

Claim 1. $F_e \in \mathcal{R}$ *for all* $e \in \mathbb{N}$.

Clearly, if there is at least one y such that $\psi_e(y) > 1$, then $F_e \in \mathcal{R}$. Now assume $\psi_e \in \mathcal{R}_{0,1}$ and suppose that there is an $x \in \mathbb{N}$ with $F_e(x)\uparrow$. Then neither an $a_s \geq x$ with $\psi_e(a_s) = 0$ nor any b_s with $\psi_e(b_s) = 1$ are found. Thus, $L = \{y \mid y \in \mathbb{N} \wedge \psi_e(y) = 0\} \supseteq B$ and L is recursive. Since A and B form a recursively inseparable pair, we conclude $|L \cap A| = \infty$ and hence there should have been an a_s such that $\psi_e(a_s) = 0$ and $a_s \geq x$. This contradiction proves Claim 1.

Claim 2. *Let* e *be any number such that* $\psi_e = \varphi_{gen_A(i)}$ *for some* $\varphi_{gen_A(i)} \in \mathcal{U}(A)$. *Then* $F_e(x) > \Phi_i(x)$ *for all* $x \in \mathbb{N}$.

First, let $F_e(x) = s$ and assume that $F_e(x)$ has been defined by finding an a_s such that $\psi_e(a_s) = 0$ and $a_s \geq x$. Since $\psi_e(a_s) = \varphi_{gen_A(i)}(a_s) = 0$, the definition of $\varphi_{gen_A(i)}$ implies both $\Phi_i \in \mathcal{R}_{mon}$ and $\Phi_i(a_s) < s$. Thus

$$F_e(x) = s > \Phi_i(a_s) \geq \Phi_i(x)$$

and we are done.

Otherwise, $F_e(x) = s$ must be defined by finding a b_s such that $\psi_e(b_s) = 1$. Since $A \cap B = \emptyset$, we have $b_s \notin A$ contradicting $\psi_e(b_s) = \varphi_{gen_A(i)}(b_s) = 1$ (cf. Definition 7). This proves Claim 2.

The rest is similar to the proof of Theorem 2, and therefore omitted.

Next, let A be simple but not hypersimple. Since A is not hypersimple, by Lemma 6, there exists a function $g \in \mathcal{R}$ with $g(x) \geq x$ for all $x \in \mathbb{N}$ such that for all but finitely many $x \in \mathbb{N}$ we have $\{x, x+1, \ldots, g(x)\} \cap \overline{A} \neq \emptyset$. Thus, in this case we define for each ψ_e a function F_e as follows.

$F_e(x) =$ "Search for the least s such that either (A) or (B) happens:
(A) $s \geq x \ \wedge \ \psi_e(y) = 1$ for all $y = s, s+1, \ldots, g(s)$
or $\psi_e(a_s) = 0$ and $a_s \geq x$,
(B) $\psi_e(s) > 1$.
If (A) happens first, then set $F_e(x) = s$.
If (B) happens first, then let $F_e(x) = 0$."

Claim 3. *$F_e \in \mathcal{R}$ for all $e \in \mathbb{N}$.*

Again, if $\psi_e \notin \mathcal{R}_{0,1}$ everything is clear. Thus, let $\psi_e \in \mathcal{R}_{0,1}$ and suppose that there is an $x \in \mathbb{N}$ with $F_e(x)\uparrow$. Then there are only finitely many $x \in A$ such that $\psi_e(x) = 0$. Hence, $\psi_e(x) = 1$ for all but finitely many $x \in A$. Moreover, $\hat{R} = \{x \mid x \in \mathbb{N} \ \wedge \ \psi_e(x) = 0\}$ is recursive and thus $R = \{x \mid x \in \mathbb{N} \setminus A \ \wedge \ \psi_e(x) = 0\}$ is also recursive because $|\{x \mid x \in A \ \wedge \ \psi_e(x) = 0\}| < \infty$. Since A is simple, $R \cap A = \emptyset$ implies that R is finite, and we conclude $\psi_e(x) = 1$ for all but finitely many $x \in \mathbb{N}$. Consequently, there must be an $s \geq x$ such that $\psi_e(y) = 1$ for all $y = s, s+1, \ldots, g(s)$, a contradiction. This proves Claim 3.

Claim 4. *Let e be any number such that $\psi_e = \varphi_{gen_A(i)}$ for some $\varphi_{gen_A(i)} \in \mathcal{U}(A)$. Then $F_e(x) > \Phi_i(x)$ for all but finitely many $x \in \mathbb{N}$.*

The case that $F_e(x) = s$ has been defined by finding $a_s \geq x$ such that $\psi_e(a_s) = 0$ can be proved as in Claim 2.

Now, suppose that there are infinitely many x such that $F_e(x) = s$ has been defined by finding an $s \geq x$ such that $\psi_e(y) = 1$ for all $y = s, s+1, \ldots, g(s)$.

Thus, by the choice of g, we conclude that there must be a $y \in \overline{A}$ such that $\psi_e(y) = 1$, again a contradiction to $\psi_e(y) = \varphi_{gen_A(i)}(y)$. This proves Claim 4.

Now, let $b(x) = \max\{F_e(x) | \ e \leq x\}$. Thus, for all $e \in \mathbb{N}$ we have $b(x) \geq F_e(x)$ for all but finitely many $x \in \mathbb{N}$. Thus, this function $b \in \mathcal{R}$ contradicts Claim 3 in the proof of Theorem 2 and Case (b) follows.

Finally, we prove Assertion (c). Assume without loss of generality that $0 \in A$ and $a_0 = 0$. Then the function $d_A(x) = \max\{s | \ a_s \leq x\}$ is total and recursive relative to A. If now $m(x) \geq d_A(x)$ for all $x \in \mathbb{N}$, then the function f_m generated by m in accordance with Definition 7 is equal to the characteristic function c_A of A, i.e., we have for

$$f_m(x) = \begin{cases} 1, & \text{if } (\exists s \leq m(x))\, [a_s = x]; \\ 0, & \text{if } (\forall s \leq m(x))\, [a_s \neq x] \end{cases}$$

the identity $f_m = c_A$.

So one can define the following A-recursive function h.

$$h(x) = \min\{y | \ y \in \mathbb{N},\ y \geq x\ \wedge\ (\forall j \leq x)\, (\exists z)\, [x \leq z \leq y\ \wedge\ \psi_j(z) \neq c_A(z)]\}$$

Since A is not recursive, no function ψ_j can be a finite variant of c_A. Hence, we can conclude that h is total, too. Moreover, by construction we have $h(x) \geq x$ for all $x \in \mathbb{N}$.

Using this function h we next define the following total A-recursive function g.

$$g(x) = \sum_{y=x}^{h(x)} d_A(y)$$

Since A is not high, there is a function $b \in \mathcal{R}$ such that

$$b(x) > g(x) \text{ for infinitely many } x\ .$$

By Claim 3 in the demonstration of Theorem 2, there exists an $e \in \mathbb{N}$ such $\Phi_e \in \mathcal{R}_{mon}$ and $\Phi_e(x) > b(x)$ for all $x \in \mathbb{N}$. Consequently,

$$\Phi_e(x) > g(x) \text{ for infinitely many } x\ . \tag{1}$$

Next, let $\psi_k \in \mathcal{R}_\psi$ be arbitrarily fixed. By (1), there exists an $x > k$ such that $\Phi_e(x) > g(x)$.

Finally, consider all $y = x,\, x+1,\, \ldots, h(x)$. Then, by the definition of g and by $\Phi_e \in \mathcal{R}_{mon}$, we have $\Phi_e(y) > d_A(y)$ for all these y. Thus, by the choice of d_A and the definition of $\varphi_{gen_A(e)}$ we arrive at

$$\varphi_{gen_A(e)}(y) = c_A(y) \text{ for all } y = x,\, x+1,\, \ldots, h(x)\ .$$

But now the definition of the function h guarantees that $\psi_k(z) \neq \varphi_{gen_A(e)}(z)$ for some z with $x \leq z \leq h(x)$. Consequently, $\varphi_{gen_A(e)}$ differs from all ψ_k in contradiction to the assumption $\mathcal{U}(A) \subseteq \mathcal{R}_\psi$. ∎

Although no complexity measure and no non-recursive recursively enumerable set is known to these authors so far, for which $\mathcal{U}(A)^{(\varphi,\Phi)}$ is in *NUM*, it is also not proven that such complexity measures do not exist for all non-recursive sets A. Nevertheless, for every non-recursive and recursively enumerable set A there is a complexity measure such that $\mathcal{U}(A)^{(\varphi,\Phi)}$ is not in *NUM*.

THEOREM 8. *For every non-recursive and recursively enumerable set A there is a complexity measure (φ, Φ) such that $\mathcal{U}(A)^{(\varphi,\Phi)} \notin NUM$.*

Proof. Let $a_0, a_1, \ldots$ be a one-one enumeration of A and let A be non-recursive. Furthermore, let $\vartheta_0, \vartheta_1, \ldots$ be an enumeration of all functions from $\mathcal{P}^2$ and let, for every k, $\vartheta_{e,k}$ denote the function $x \to \vartheta_e(k,x)$, that is, the k-th function in the numbering ϑ_e.

The main idea of the proof is now to define a numbering ψ such that ψ_e generates a function outside the family ϑ_e whenever $\vartheta_e \in \mathcal{R}^2$. Then using Lemma 1 one adds this family to some given complexity measure and obtains, that the resulting complexity measure (φ, Φ) defines a class $\mathcal{U}(A)^{(\varphi,\Phi)}$ outside *NUM*. The numbering ψ is constructed as follows:

Algorithm. "For each e, define inductively for $k = 0, 1, \ldots$ numbers $x_{e,k}$ and $s_{e,k}$ such that the following holds:

- $a_{s_{e,k}} = x_{e,k}$ and $s_{e,k} > x_{e,k}$;
- $s_{e,k}$ is so large that $\vartheta_{e,k}(x_{e,k})$ is computed in less than $s_{e,k}$ computational steps;
- $x_{e,k} > s_{e,k'}$ for all $k' < k$.

If the search does not terminate for e, k, then the numbers $x_{e,k'}, s_{e,k'}$ with $k' \geq k$ do not exist. Furthermore, let

$$\psi_e(x) = \begin{cases} s_{e,k}+1, & \text{if } \vartheta_{e,k}(x_{e,k})\!\downarrow = 0 \text{ where } k \text{ is the first number found} \\ & \text{such that } x_{e,k} \text{ exists and } x_{e,k} \geq x; \\ s_{e,k}-1, & \text{if } \vartheta_{e,k}(x_{e,k})\!\downarrow > 0 \text{ where } k \text{ is the first number found} \\ & \text{such that } x_{e,k} \text{ exists and } x_{e,k} \geq x; \\ \uparrow, & \text{otherwise, i.e., there is no } k \text{ where} \\ & x_{e,k} \text{ exists, } x_{e,k} \geq x \text{ and } \vartheta_{e,k}(x_{e,k})\!\downarrow.\text{"} \end{cases}$$

Claim 1. *If $\vartheta_e \in \mathcal{R}^2$ then the search terminates for all k, that is, all $x_{e,k}$ and $s_{e,k}$ exist.*

Assume by way of contradiction that $\vartheta_e \in \mathcal{R}^2$ and k is the first index such that $x_{e,k}$ and $s_{e,k}$ are not defined. Assume furthermore that $x \in A$ and $x > s_{e,k'}$ for all $k' < k$. Let $y = \max(\{0\} \cup \{s_{e,k'} \mid k' < k\})$ and let the function ct be defined such that $ct(x)$ be the number of computational steps to compute $\vartheta_{e,k}(x)$. ct is a total recursive function.

Given $x > y$ the x does not qualify to become $x_{e,k}$. Thus either $x \notin A$ or $x = a_s$ for some $s < ct(x)$. Thus $x \in A$ iff $x = a_s$ for some $s < ct(x)$ and A would be recursive in contradiction to the choice of A. This completes the proof of Claim 1.

Claim 2. *The set* $\{(e, x, \psi_e(x)) \mid \psi_e(x)\downarrow\}$ *is recursive.*

It is easy to see that the ψ_e are uniformly partial recursive. Furthermore, $\psi_e(x)\downarrow \leq t$ can only happen if the above search finds a k with $x \leq x_{e,k} \leq t+2$ and $s_{e,k} \leq t + 2$. Thus one can bound the search by a polynomial in $e + t$ and therefore the uniform graph $\{(e, x, t) \mid \psi_e(x)\downarrow \leq t\}$ is recursive. This completes the proof of Claim 2.

So it is legal to apply Lemma 1. Furthermore, it is immediate from the definition of ψ_e that ψ_e generates a function different from all $\vartheta_{e,k}$ whenever $\vartheta_e \in \mathcal{R}^2$. Therefore, all possible witnesses to show that $\mathcal{U}(A)^{(\varphi,\Phi)}$ is in NUM are diagonalized. ∎

5 Reliable and *EX*-Learnability of $\mathcal{U}(A)$

Blum and Blum [6] showed that $\mathcal{B}$ is in $\mathcal{R}EX$. The *EX*-learnability of $\mathcal{U}(A)$ alone can be generalized to every r.e. set A, but this is not possible for reliability. But before dealing with reliable *EX*-inference, we show that every $\mathcal{U}(A)$ is *EX*-learnable.

THEOREM 9. *The class* $\mathcal{U}(A)$ *is EX-learnable for all r.e. sets* A.

Proof. If A is recursive, then $\mathcal{U}(A) \in NUM$ (cf. Theorem 5) and thus *EX*-learnable. So let A be non-recursive and let $a_0, a_1, \ldots$ be the fixed recursive one-one enumeration of A. Next, we define the desired learner M. Let $f \in \mathcal{U}(A)$ and let $n \in \mathbb{N}$.

$M(f^n) =$ "Delete all $e \leq n$ such that there is an $x \leq n$ satisfying one of the following three conditions:

(a) $\Phi_{gen_A(e)}(x) \leq n$ and $\varphi_{gen_A(e)}(x) \neq f(x)$,
(b) $f(x) = 0$, $\neg[\Phi_e(x) \leq n]$ and there is an $s \leq n$ such that $a_s = x$,
(c) $\Phi_e(x+1) \leq n$ and $\neg[\Phi_e(x) \leq \Phi_e(x+1)]$.

If all $e \leq n$ have been deleted then output $gen_A(n+1)$. Otherwise, output $gen_A(e)$ for the least non-deleted $e \leq n$."

Since the set $\{e, x, n \mid e, x, n \in \mathbb{N} \text{ and } \Phi_e(x) \leq n\}$ is recursive and since $a_0, a_1, \ldots$ is a recursive enumeration of A, we directly obtain $M \in \mathcal{R}$.

It remains to show that M does EX-learn $\mathcal{U}(A)$. By construction, no index e with $\varphi_{gen_A(e)} = f$ can be deleted. Now, let i be the least correct index for f, i.e., $\varphi_{gen_A(i)} = f$. It remains to show that every $e \leq i$ is eventually deleted. This is obvious if there exists an x such that $\varphi_{gen_A(e)}(x)\downarrow$ and $\varphi_{gen_A(e)}(x) \neq f(x)$. Otherwise, let z be the least argument x such that $\varphi_{gen_A(e)}(x)\uparrow$. Now, $\varphi_{gen_A(e)}(x)\uparrow$ either because of $\Phi_e(z)\downarrow$ and there is an $x < z$ such that $\neg[\Phi_e(x) \leq \Phi_e(x+1)\downarrow]$ or since $\Phi_e(z)\uparrow$. In the first case, e is eventually deleted by checking Condition (c). In the second case, there is either a $y > z$ such that $\Phi_e(y)\downarrow$, then e is again deleted by Condition (c). Otherwise, $\Phi_e(y)\uparrow$ for all $y > z$, too. Now, since A is not recursive there must be an $x \geq z$ and an s such that $x = a_s$ and $f(x) = 0$. For seeing this, it suffices to note that $f(x) = 0$ for all $f \in \mathcal{U}(A)$ and all $x \in \mathbb{N} \setminus A$. Thus, if $f(x) = 1$ for all but finitely many $x \in A$ then A would be recursive, a contradiction. Thus, Condition (b) must happen eventually. ∎

The result that $\mathcal{B}$ is reliably EX-learnable can be generalized to halves of recursively inseparable pairs and to simple but not hypersimple sets.

THEOREM 10. *$\mathcal{U}(A)$ is reliably EX-learnable if*
(a) *A is part of a recursively inseparable pair, or*
(b) *A is simple but not hypersimple.*

Proof. Again let $a_0, a_1, \ldots$ be a one-one enumeration of A. Recall that $gen_A(e)$ is the program computing the function $\varphi_{gen_A(e)}$ associated to Φ_e.

The central idea of the proof is that conditions (a) and (b) allow to identify a class of functions which contains all recursive functions which are too difficult to learn and on which the learner then diverges. The recursive functions outside these class turn out to be EX-learnable and contain the class $\mathcal{U}(A)$.

The learner M does not need to succeed on functions f where the range of f contains values outside $\{0, 1\}$ or if $f(x) = 1$ for all but finitely many $x \in A$. The second condition cannot be checked effectively, but, for recursive functions f and the A in the precondition of the theorem, one can check it indirectly as follows:

In Case (a), let $B = \{b_0, b_1, \ldots\}$ form together with A a recursively inseparable pair. If $f(x) = 1$ for almost all $x \in A$ then $f(b_s) = 1$ for some b_s. So one defines that f^n disqualifies if $f(x) > 1$ for some $x \leq n$ or if $f(b_s) = 1$

for some $s \leq n$ and $b_s \leq n$.

In Case (b), the set A is simple but not hypersimple. By Lemma 6 there is a function $g \in \mathcal{R}$ with $g(x) \geq x$ for all $x \in \mathbb{N}$ such that $\overline{A}$ intersects every interval $\{x, x+1, \ldots, g(x)\}$. But if $f(x) = 1$ for almost all $x \in A$, then, by the simplicity of A, $f(x) = 1$ for almost all $x \in \mathbb{N}$. Hence, there must be an x such that $f(y) = 1$ for all $y \in \{x, x+1, \ldots, g(x)\}$. So one defines that f^n disqualifies if $f(x) > 1$ for some $x \leq n$ or if there exists an $x \leq n$ such that $g(x) \leq n$ and $f(y) = 1$ for all $y \in \{x, x+1, \ldots, g(x)\}$.

The reliable EX-learner N is a modification of the learner M from Theorem 9 which copies M on all f^n except on those which disqualify — on them, N always outputs the canonical index of $f^n 2^\infty$ and thus either converges to some $\sigma 2^\infty$ or diverges by changing its hypothesis infinitely often. Let $c(f^n)$ be the canonical index for $f^n 2^\infty$.

$$N(f^n) = \begin{cases} c(f^n), & \text{if } f^n \text{ is disqualified;} \\ M(f^n), & \text{otherwise.} \end{cases}$$

For the verification, one should note that every $f \in \mathcal{U}(A)$ takes the value 0 on all arguments $x \in \overline{A}$. Thus, no f^n is disqualified for all $f \in \mathcal{U}(A)$ and therefore N is an EX-learner for $\mathcal{U}(A)$.

Now, let $f \in \mathcal{R}$ and assume that $\{N(f^n)\}_{n \in \mathbb{N}}$ converges to some i. We have to show that $\varphi_i = f$.

First note that if f^n is disqualified for some $n \in \mathbb{N}$ then every f^{n+m} is disqualified, too, for all $m \in \mathbb{N}$. Thus, if convergence happens since for some $n \in \mathbb{N}$ the initial segment f^n is disqualified, then $f = f^n 2^\infty$ and we are done.

Otherwise, no f^n is disqualified and $N(f^n) = M(f^n)$ for all $n \in \mathbb{N}$. Since M outputs exclusively indices of the form $gen_A(e)$ there must be a least e such that $i = gen_A(e)$. Let us first assume that $f(x) = 0$ for infinitely many $x \in A$. Then, as the proof of Theorem 9 shows, M converges only in case if $\varphi_{gen_A(e)} = f$.

Finally, suppose $f(x) = 0$ for only finitely many $x \in A$. Then, in Case (a) we can conclude that $f(x) = 1$ for some $x \in B$, and in Case (b) that there must be some x such that $f(y) = 1$ for all $y = x, x+1, \ldots, g(x)$. In both cases, some f^n would have been disqualified, and hence this case cannot occur. ∎

Next, we ask what happens if A is hypersimple. The answer is provided by the following theorem.

Theorem 11. *If A is hypersimple and not high then $\mathcal{U}(A)$ is not reliably*

EX-learnable.

Proof. Let $a_0, a_1, \dots$ be a one-one enumeration of the hypersimple set A and assume that $\mathcal{U}(A)$ is reliably EX-learnable. Then also the union

$$\mathcal{U}(A) \cup \{\sigma 1^\infty \,|\, \sigma \in \{0,1\}^*\}$$

is reliably EX-learnable, since every class in NUM is reliably EX-learnable and the reliably EX-learnable classes are closed under union (cf. [6,27]). Let M be any given EX-learner for the above union. Now, we define the following function h_1 by taking $h_1(x)$ as

$$\begin{aligned} h_1(x) \;=\; & \min\{s \geq x \,|\, (\forall \sigma \in \{0,1\}^x)\,(\forall n \leq x)\,[M(\sigma 1^s) \neq M(\sigma) \vee \\ & \Phi_{M(\sigma)}(n) \leq s \;\wedge\; \varphi_{M(\sigma)}(n) = \sigma(n)]\}. \end{aligned}$$

The function h_1 is total since every output $M(\sigma)$ either computes the function $\sigma 1^\infty$ or $\varphi_{M(\sigma)} \neq \sigma 1^\infty$ and thus there must be an s such that $M(\sigma) \neq M(\sigma 1^s)$. Moreover, by the definition of h_1 we may conclude that $h_1(x) \geq x$ for all $x \in \mathbb{N}$. Without loss of generality we may also assume $h_1 \in \mathcal{R}_{mon}$ for otherwise, we could take $\hat{h}_1$ defined as $\hat{h}_1(0) = h_1(0)$ and for all $x > 0$, $\hat{h}_1(x) = h_1(x) + \hat{h}_1(x-1)$.

Using the function h_1 defined above, one can define a function $h_2(x)$ as

$$\begin{aligned} h_2(x) \;=\; & \min\{s \,|\, (\exists y)\,[x \leq y \leq s \\ & \wedge\, y,\, y+1,\, \dots,\, y + h_1(y) \in \{a_0,\, a_1,\, \dots,\, a_s\} \\ & \wedge\, h_1(y + h_1(y)) < s]\}. \end{aligned}$$

Since A is hypersimple, we directly get from Lemma 6 that $h_2 \in \mathcal{R}$.

Finally, let κ be a recursive function such that for all $i, j, x \in \mathbb{N}$ and all $\tau \in \{0,1\}^*$

$$\varphi_{\kappa(i,j,\tau)}(x) = \begin{cases} \tau(x), & \text{if } x \in dom(\tau); \\ \varphi_i(x), & \text{if } \Phi_i(x) \leq h_1(h_2(x)); \\ 1, & \text{if } a_t = x \text{ and } t \leq \Phi_j(x) \leq h_2(x) \text{ for some } t; \\ 0, & \text{otherwise.} \end{cases}$$

In the above definition of $\varphi_{\kappa(i,j,\tau)}(x)$, the first case that is applicable is always taken. Since the search-conditions in the second and third case are bounded by a recursive function in x, the family of all $\varphi_{\kappa(i,j,\tau)}$ contains only recursive functions and its universal $i, j, \tau, x \to \varphi_{\kappa(i,j,\tau)}(x)$ is computable in all parameters. Thus, for finishing the proof it suffices to show that $\mathcal{U}(A) \subseteq \{\varphi_{\kappa(i,j,\tau)} \,|\, i, j, x \in \mathbb{N} \;\wedge\; \tau \in \{0,1\}^*\}$. Then we could conclude $\mathcal{U}(A) \in NUM$, a contradiction to Theorem 7, since A is neither recursive nor high.

Consider for every $f \in \mathcal{U}(A)$ two particular indices, i.e., an index j with $f = \varphi_{gen_A(j)}$ and the index i to which $(M(f^n))_{n\in\mathbb{N}}$ converges. Let z be the least n such that $M(f^{n+m}) = i$ for all $m \in \mathbb{N}$ and let $\tau = f(0), f(1), \dots, f(z)$.

Claim. $f = \varphi_{\kappa(i,j,\tau)}$

Clearly, if $x \leq z$ then $\varphi_{\kappa(i,j,\tau)}(x) = \tau(x) = f(x)$ by Case 1 of the definition of $\varphi_{\kappa(i,j,\tau)}$, and we are done.

Next, let $x > z$ be arbitrarily fixed. Consider the y from the definition of $h_2(x)$ and let $\sigma = f(0), f(1), \dots, f(y)$. Next, we look at the computation of $h_1(y)$.

First, assume that the computation of $h_1(y) = s$ has terminated by verifying $\Phi_{M(\sigma)}(n) \leq s$ and $\varphi_{M(\sigma)}(n) = \sigma(n)$ for all $n \leq y$. By construction, $i = M(\tau) = M(\sigma)$, and since M EX-learns $\mathcal{U}(A)$, we also conclude $\varphi_i = f$. Thus, it has been verified that $\Phi_i(n) \leq s$ for all $n \leq y$. Since $x \leq y \leq h_2(x)$, we have in particular that $\Phi_i(x) \leq s = h_1(y)$ and since $h_1 \in \mathcal{R}_{mon}$, this implies $\Phi_i(x) \leq h_1(h_2(x))$. Consequently, Case 2 in the definition of $\varphi_{\kappa(i,j,\tau)}$ is applicable and therefore $\varphi_{\kappa(i,j,\tau)}(x) = \varphi_i(x) = f(x)$.

Otherwise, $M(\sigma 1^{h_1(y)}) \neq M(\sigma)$ has been verified during the computation of $h_1(y)$. Since $M(\tau) = M(\sigma) = i$, there must be some $a_{t'} \in \{y, y+1, \dots, y+h_1(y)\}$ with $f(a_{t'}) = 0$. As a consequence, $\Phi_j(a_{t'}) < t'$, by the definition of $\mathcal{U}(A)$. Furthermore, $x \leq y$ as well as $y, y+1, \dots, y+h_1(y) \in \{a_0, a_1, \dots, a_s\}$ and $\Phi_j \in \mathcal{R}_{mon}$ imply that $t' \leq s$ and $\Phi_j(x) \leq \Phi_j(a_{t'}) < t \leq h_2(x)$.

Thus, if $x = a_t$ then Case 3 in the definition of $\varphi_{\kappa(i,j,\tau)}$ is not applicable and it holds that $t > \Phi_j(a_t)$. It follows that $\varphi_{\kappa(i,j,\tau)}(x) = 0$ and $f(x) = \varphi_{gen_A(j)}(x) = 0$.

Finally, if Case 3 is applicable then the choice of j and the definition of $\mathcal{U}(A)$ imply that $f(x) = \varphi_{gen_A(j)}(x) = 1$, and we are again done. ∎

Next, we turn our attention to the problem whether or not the results obtained so far are generalizable to the larger classes $\tilde{\mathcal{B}}$ and $\tilde{\mathcal{U}}(A)$ obtained by replacing $\Phi_i \in \mathcal{R}_{mon}$ by $\Phi_i \in \mathcal{R}$ in Definitions of the classes $\mathcal{B}$ and $\mathcal{U}(A)$, respectively. We shall refer to these classes as non-monotone approximations.

6 Non-Monotone Approximations

Blum and Blum [6] defined their class $\mathcal{B}$ using only monotone complexity functions. However, no motivation for this condition has been provided in [6] except that it was needed to prove $\mathcal{B} \in \mathcal{R}EX$. Therefore, we ask whether or not this condition $\Phi_i \in \mathcal{R}_{mon}$ is necessary to ensure the learnability results obtained so far. That is, instead of $\mathcal{B}$, we now consider the class

$$\tilde{\mathcal{B}}^{(\varphi,\Phi)} = \{\varphi_{\tau(i)} |\ i \in \mathbb{N} \text{ and } \Phi_i \in \mathcal{R}\}$$

where the $\varphi_{\tau(i)}$ depend on the Φ_i in the same way as in the case of $\mathcal{B}$:

$$\varphi_{\tau(i)}(x) = \begin{cases} 1, & \text{if } \Phi_i(x)\downarrow \text{ and } \Phi_x(x) \leq \Phi_i(x), \\ 0, & \text{if } \Phi_i(x)\downarrow \text{ and } \neg[\Phi_x(x) \leq \Phi_i(x)], \\ \uparrow, & \text{otherwise.} \end{cases}$$

In contrast to our results obtained so far, now the answer depends on the complexity function $\Phi \in \mathcal{P}^2$ involved. As a matter of fact, one can play a kind of coding game by suitably modifying Φ as the proof of the following theorem shows.

THEOREM 12. *For every Gödel numbering* φ *there exists a complexity function* $\Phi \in \mathcal{P}^2$ *such that* $\tilde{\mathcal{B}}^{(\varphi,\Phi)} \in EX$.

Proof. Let φ be any Gödel numbering and let $\hat{\Phi} \in \mathcal{P}^2$ be any complexity function for φ. We define $\Phi \in \mathcal{P}^2$ by setting $\Phi_i(x) = \hat{\Phi}_i(x) + i$ for all $i, x \in \mathbb{N}$. Note that the result of the addition is not defined if $\hat{\Phi}_i(x)\uparrow$. Consequently, $dom(\varphi_i) = dom(\Phi_i)$ for all $i \in \mathbb{N}$. Furthermore, the predicate $\{(i,x,y) |\ i,x,y \in \mathbb{N},\ \Phi_i(x) \leq y\}$ can be uniformly decided by testing $\hat{\Phi}_i(x) + i \leq y$. Thus, Φ is a complexity function.

Claim. $\tilde{\mathcal{B}}^{(\varphi,\Phi)} \in EX$.

The main idea for proving the claim is to reduce the EX-learning problem to learning with additional information as defined in Freivalds and Wiehagen [11]. That means, the desired EX-learner first aims to find a number t such that at least one of the functions $\varphi_{\tau(0)}, \dots, \varphi_{\tau(t)}$ correctly computes the target function $f \in \tilde{\mathcal{B}}^{(\varphi,\Phi)}$. When such a number t has been found, it simply simulates the machine given in [11] on input f^n and t by using the hypothesis space $(\varphi_{\tau(i)})_{i \in \mathbb{N}}$. Since this machine correctly learns f the claim then follows.

Next, we describe how such a number t can be found. Let $f \in \tilde{\mathcal{B}}^{(\varphi,\Phi)}$ be any target function, and let $n \in \mathbb{N}$. Then the EX-learner M simulates the computation of $\Phi_0(0), \Phi_1(1), \dots, \Phi_n(n)$ for n steps. For every $\Phi_j(j)$, $0 \leq j \leq n$, that turned out to be defined within n steps of computation, M now checks whether or not $f(j) = 0$. Assuming such a j has been found, M computes $t = \Phi_j(j)$.

It remains to show that t has the property described above and that the search for t always succeeds. For the first part, assume any i such that

$\varphi_{\tau(i)} = f$. Since $f(j) = 0$, the definition of $\tilde{\mathcal{B}}^{(\varphi,\Phi)}$ and the complexity measure Φ directly yields

$$t = \Phi_j(j) > \Phi_i(j) \geq i \ .$$

Hence, at least one of the functions $(\varphi_{\tau(i)})_{i \in \mathbb{N}}$ must compute f .

The second part directly follows from the non-recursiveness of the characteristic function c_K of K . By construction, $\varphi_{\tau(j)}(x) \leq c_K(x)$ for all $j, x \in \mathbb{N}$, and since $c_K \notin \mathcal{R}$ for every j there must be an x such that $0 = \varphi_{\tau(j)}(x) < c_K(x) = 1$. Thus, M 's search for t always succeeds. This proves the Claim, and thus the theorem is shown. ∎

However, the situation changes considerably for more natural complexity functions. A complexity measure (φ, Φ) is said to be *interpolating* if its complexity function Φ satisfies the following condition. For every $i \in \mathbb{N}$ and for every string $\sigma \in \mathbb{N}^*$, there is an index e with

$$\Phi_e(x) = \begin{cases} \sigma(x), & \text{if } x \in dom(\sigma); \\ \Phi_i(x), & \text{if } x \in dom(\Phi_i) \setminus dom(\sigma); \\ \uparrow, & \text{otherwise.} \end{cases}$$

For example, let φ be the canonical Gödel numbering of all 3-tape Turing machine (with input tape, work tape and output tape), and let the complexity function Φ be defined by assigning for every $i, x \in \mathbb{N}$ to $\Phi_i(x)$ the amount of space used on the work tape while computing $\varphi_i(x)$. Then it is easy to see that (φ, Φ) is interpolating.

However, for proving our negative result concerning the learnability of $\tilde{\mathcal{B}}^{(\varphi,\Phi)}$ we need a further technical condition which we define next.

DEFINITION 8. A complexity measure (φ, Φ) is *regular* iff it is interpolating and $\Phi \in \mathcal{P}^2$ additionally satisfies the following property.

$$(\forall p \in \mathcal{R})\,(\forall i,\ \varphi_i = p)\ \ [\ \ (\forall c \in \mathbb{N})\,(\forall \psi \in \mathcal{P})\,(\forall k,\ \varphi_k = \psi)\,(\exists j \in \mathbb{N})\,(\forall x)$$
$$\varphi_j(x) \ = \ \begin{cases} \psi(x), & \text{if } \ p(x) = c; \\ 0, & \text{if } \ p(x) \neq c; \end{cases}$$
$$\Phi_j(x) \ = \ \begin{cases} \max\{\Phi_i(x), \Phi_k(x)\}, & \text{if } \ p(x) = c; \\ \Phi_i(x), & \text{if } \ p(x) \neq c \ \ ] \end{cases}$$

where j can be effectively computed from i and k and where, by convention, $\max\{\Phi_i(x), \Phi_k(x)\} = \infty$ if $\Phi_k(x) = \infty$, that is, if $\varphi_k(x)$ is not defined.

In other words, an interpolating measure is regular iff it satisfies the additional property, that computing a function under the condition that $p(x) = c$ is on those x, where the condition is not satisfied, only as complex as the computation of the parameter $p(x)$ of the test-condition. In particular, this complexity is independent of the value of the constant c and from the program of the function to be computed under the condition that $p(x) = c$.

Note that the space complexity measure (φ, Φ) defined above is a regular complexity measure while the measure adding the size of the Turing table to such a resource is neither regular nor interpolating.

Now, we are ready to show that the condition $\Phi_i \in \mathcal{R}_{mon}$ in the definition of the class $\mathcal{B}$ cannot be dropped without loosing $\mathcal{R}EX$-learnability, since our next theorem shows that the class $\tilde{\mathcal{B}}^{(\varphi,\Phi)}$ is in general even not BC-learnable. Bārzdiņš and Podnieks — see [8] for more details — showed that BC is equivalent to correctly predicting the functions to be learned almost everywhere. Thus, the following proof shows that, for every partial recursive M_k, there is a function $g \in \tilde{\mathcal{B}}^{(\varphi,\Phi)}$ which is infinitely often incorrectly predicted by M_k.

THEOREM 13. *$\tilde{\mathcal{B}}^{(\varphi,\Phi)}$ is not BC-learnable for every regular complexity measure (φ, Φ).*

Proof. Let $M_1, M_2, \ldots$ be the canonical enumeration of all partial prediction machines [18]. In the following we enumerate uniformly for every $e > 0$ a set A_e such that M_e fails to predict the characteristic function c_{A_e} of A_e and that there is a j such that Φ_j is total and $A_e = \{z \mid \Phi_z(z) \leq \Phi_j(z)\}$. Using the Recursion Theorem, we can find a one-one increasing function h such that A_e is of the form $(K \cap \{h(e,0), h(e,1), h(e,2), \ldots\}) \cup \{z \mid \Phi_z(z) \leq \Phi_i(z)\}$ where φ_i, Φ_i are defined simultaneously with h below. This simultaneous definition needs the following function p (also depending on h):

$$p(z) = \begin{cases} e, & \text{if } e > 0 \text{ and } h(e,x) = z \text{ for some } x \in \mathbb{N}; \\ 0 & \text{otherwise.} \end{cases}$$

Let $i \in \mathbb{N}$ be the canonical program for p computed according the above definition from a program for h. For every $e > 0$, let A_e contain all z with $\Phi_z(z) \leq \Phi_i(z)$ and all $h(e, x_l)$ where x_l is defined in the below enumeration process (with g_l being an approximation to c_{A_e}).

"For every $l \in \mathbb{N}$ and all $z \in \mathbb{N}$ let

$$g_l(z) = \begin{cases} 1, & \text{if } z = h(e, x_{l'}) \text{ for some } l' < l \text{ or } \Phi_z(z) \leq \Phi_i(z); \\ 0, & \text{otherwise.} \end{cases}$$

Search in parallel until an $x' > x_{l-1}$ ($x' \geq 0$ in case $l = 0$) is found such that $M_e(g_l^{h(e,x')-1}) \downarrow = 0$ and let $x_l = x'$ in the case where the search has terminated."

Note that both, K and A_e, contain exactly those elements of the form $h(e,x)$ where x equals some x_l defined in the above enumeration process. Furthermore, the set A_e is a recursive subset of K and so there is a function $\varphi_k \in \mathcal{R}$ such that Φ_k satisfies $\Phi_k(z) \geq \Phi_z(z)$ for all $z \in A_e$. Since (φ, Φ) is regular, there is a complexity function Φ_j (of a suitable φ_j) such that

$$\Phi_j(z) = \begin{cases} \max\{\Phi_i(z), \Phi_k(z)\} & \text{if } p(z) = e; \\ \Phi_i(z) & \text{if } p(z) \neq e. \end{cases}$$

It follows that the function $g = \varphi_{\tau(j)}$ coincides with the characteristic function c_{A_e} of A_e. Therefore $c_{A_e} \in \tilde{\mathcal{B}}^{(\varphi,\Phi)}$. It remains to show that M_e does not predict g, which then implies that $\tilde{\mathcal{B}}^{(\varphi,\Phi)}$ is not BC-learnable.

Case 1. The sequence of the $x_0, x_1, \ldots$ is finite, that is, there is a number l such that the $x_{l'}$ with $l' < l$ are found but the search for x_l does not terminate.

Then $g = g_l$ and the learner M_e fails to predict $g(h(e,x))$ for all x with $x > x_{l'}$ for all $l' < l$. In particular, M_e fails to predict the value $g(z)$ for infinitely many z.

Case 2. All x_l are defined.

For every l, the functions g and g_l coincide below $h(e,x_l)$. In particular, M_e predicts 0 at every input $g^{h(e,x_l)-1}$ although $g(h(k,x_l)) = 1$. So M_e makes infinitely many wrong predictions for the function g. ∎

Let us now consider the counterpart of $\mathcal{U}^{(\varphi,\Phi)}(A)$ defined by replacing the condition $\Phi_e \in \mathcal{R}_{mon}$ by $\Phi_e \in \mathcal{R}$, i.e.,

$$\tilde{\mathcal{U}}(A)^{(\varphi,\Phi)} = \{f_{gen_A(e)} \mid \Phi_e \in \mathcal{R}\} .$$

The learning algorithm in Theorem 12 can be easily adapted to work for every recursively enumerable set A in place of K. Thus, we immediately obtain the following result.

COROLLARY 14. *For every Gödel numbering φ there exists a complexity function $\Phi \in \mathcal{P}^2$ such that, for any recursively enumerable set A, the class $\tilde{\mathcal{U}}(A)^{(\varphi,\Phi)}$ is EX-learnable.*

The next result shows that there are also complexity measures (φ, Φ) for which the class $\tilde{\mathcal{U}}(A)^{(\varphi,\Phi)}$ is even not BC-learnable. Note that the measure is less natural than the one in Theorem 13 since in Theorem 13 the non-learnability holds for every regular measure and thus for a measure as natural as the space-complexity of a three-tape Turing machine.

THEOREM 15. *For every non-recursive but recursively enumerable set* A *there is a complexity measure* (φ, Φ) *such that* $\tilde{\mathcal{U}}(A)^{(\varphi,\Phi)}$ *is not BC-learnable.*

Proof. Let $a_0, a_1, \ldots$ be the one-one enumeration of the given set A. Using Lemma 1 one can add to a given complexity measure all primitive recursive functions and also the functions ψ_e defined below. Let $M_0, M_1, \ldots$ be a canonic enumeration of all prediction-learners [18].

"For each $e \in \mathbb{N}$, define the function ψ_e in stages $s = 0, 1, \ldots$; enumerate an auxiliary set B_e initialized as $\emptyset$ and start with ψ_e being everywhere undefined.

In stage 0, let $\psi_e(a_0) = 0$ and let $B_e = \{a_0\}$.

In stage s, $s \geq 1$, do the following provided that $\psi_e(a_s)$ is still undefined; otherwise do nothing.

(1) Let g_s be the characteristic function of B_e before stage s.
(2) Check whether $M_e(g_s^{a_s - 1})\downarrow = 0$ within s steps.
(3) If so, then enumerate a_s into B_e and let $\psi_e(x) = s$ for all those $x \leq a_s$ for which $\psi_e(x)$ is not already defined.
(4) Otherwise, let $\psi_e(a_s) = s - 1$ and let B_e be unchanged."

Claim 1. *The predicate "*$\psi_e(x)\downarrow \leq t$*" is uniformly decidable for all* $e, x, t \in \mathbb{N}$.

Run the above algorithm for the first $t+1$ stages. Suppose that $\psi_e(x)$ does not turn out to be defined after the first $t+1$ stages. Then it takes either the value s or the value $s-1$ for some stage $s > t+1$ or it remains undefined. Thus $\psi_e(x)$ is either undefined or greater than t.

Claim 2. B_e *contains exactly the* a_s *with* $s \leq \psi_e(a_s)$.

Assume that this would be false for some a_s but true for all $a_{s'}$ with $s' < s$. If $\psi_e(a_s)$ is defined in some stage $s' < s$ then $\psi_e(a_s) = s'$ and this happened in step (3) of stage s' and a_s is not enumerated into B_e since $a_s \neq a_{s'}$. If $\psi_e(a_s)$ is defined in stage s then a_s is put into B_e iff $\psi_e(a_s)$ is defined to be s. Since ψ_e is not defined after stage s, this completes the case distinction.

Claim 3. *There is no machine* M_e *that predicts all functions in* $\tilde{\mathcal{U}}(A)^{(\varphi,\Phi)}$ *almost everywhere correctly.*

If B_e is finite then the characteristic function c_{B_e} of B_e is primitive recursive. Since furthermore $B_e \subseteq A$, there is a primitive recursive function which generates c_{B_e}. Consequently, $c_{B_e} \in \tilde{\mathcal{U}}(A)^{(\varphi,\Phi)}$ and M_e has almost everywhere to predict correctly the value 0 (cf. [8]). So there is some value s such that all elements are already enumerated into B_e at stage s and no $a_{s'}$ with $s' > s$

is below an element in B_e. It follows that the algorithm never goes through step (3) in a stage $s' > s$ and that $M_e(c_{B_e}^{a_{s'}-1})$ is undefined at stage s'. Since almost all predictions $M_e(c_{B_e}^n)$ are defined, one can use the computation time of this recursive function in order to compute for almost all $x \in A$ an upper bound for the s with $x = a_s$ — but since A is not recursive, this is impossible.

So B_e must be infinite. It follows that ψ_e is total and $\psi_e = c_{B_e}$, again $c_{B_e} \in \tilde{\mathcal{U}}(A)^{(\varphi,\Phi)}$. Each $x \in B_e$ is selected such that $M_e(c_{B_e}^{x-1})$ predicts 0, thus M_e makes infinitely many wrong predictions and M_e fails to predict c_{B_e} also in this case. This finishes the proof of Claim 3 and thus $\tilde{\mathcal{U}}(A)^{(\varphi,\Phi)} \notin BC$.
∎

7 Robust Learning

A mathematical elegant proof method to separate learning criteria is the use of classes of self-describing functions. Somehow these examples are a bit artificial, since they use coding tricks. On the other hand, as pointed out in Jain, Smith and Wiehagen [19], one could argue that self-description is quite a natural phenomenon in that every cell of every organism contains a description of itself. Nevertheless, from a learning theoretical point some criticism remains in order, since a learner needs only to fetch some code from the input. Thus, the learnability result in the separation proof is usually trivial, while the non-learnability result shows that at least one function from the separating class is too complex to gain the information necessary to learn it in the stronger model. However, if such self-describing function classes would be the only separating examples for a variety of learning models then this would clearly have major consequences for our understanding of learning and the value of the theory.

But how can one attack this problem? Bārzdiņš suggested to prove or to disprove the following conjecture.

Let $U \subseteq \mathcal{R}$; then $\Theta(U) \in EX$ for all effective operators Θ with $\Theta(U) \subseteq \mathcal{R}$ implies that there is a $\psi \in \mathcal{R}^2$ such that $U = \mathcal{R}_\psi$.

Kurtz and Smith [24] disproved Bārzdiņš' conjecture for classes $U \in NUM$. Jain, Smith and Wiehagen [19, Section 6] disproved Bārzdiņš's conjecture by giving a class U containg 0^∞ and all total functions φ_e which extend 0^e1. Furthermore, many interesting variants of these results have been stated (cf., e.g., [1,7,12,17,19,29,34]).

In particular, Fulk [12] proposed the notion of *robust* learnability. For example, a class U is robustly EX-learnable, iff, for every operator Θ, the class

$\Theta(U)$ is EX-learnable. There were many discussions, which operators Θ are admissible in this context. Fulk [12] considered the class of all general recursive operators Θ which map every total function to a total one. For this version, Fulk [12] disproved Bārzdiņš' conjecture, and Jain *et al.* [19] provided a class of self-describing functions which is robustly EX-learnable.

However, almost nothing has been known concerning the robust EX-learnability of "naturally defined function classes" such as $\mathcal{B}^{(\varphi,\Phi)}$ and $\mathcal{U}(A)^{(\varphi,\Phi)}$. We therefore address this problem here. First, we consider $\mathcal{B}^{(\varphi,\Phi)}$. Again, the answer depends on the complexity measure involved. If we take any complexity function $\Phi \in \mathcal{P}^2$ such that $\Phi_i(x) \geq i$ for all $i, x \in \mathbb{N}$, i.e., any complexity measure similar to the one defined in the proof of Theorem 12, then this function provides the information needed to robustly EX-identify $\mathcal{B}^{(\varphi,\Phi)}$ by methods similar to those of Jain, Smith and Wiehagen [19]. Therefore, we shall refer to these complexity functions as to *coding complexity functions*.

THEOREM 16. *For every Gödel numbering φ and every associated coding complexity function $\Phi \in \mathcal{P}^2$ the class $\mathcal{B}^{(\varphi,\Phi)}$ is robustly EX-learnable.*

Proof. Let Θ be any general recursive operator. Then Θ coincides on $\mathcal{R}$ with a total effective operator [5] Ξ, i.e., $\Theta(f) = \Xi(f)$ for all $\varphi \in \mathcal{R}$. Let $\alpha \in \mathcal{R}$ be such that $\Xi(\varphi_i) = \varphi_{\alpha(i)}$ for all $i \in \mathbb{N}$. Furthermore, as usual we use c_K to denote the characteristic function of K.

We have to construct a machine M that EX-learns $\Theta(\mathcal{B}^{(\varphi,\Phi)})$. For doing this, we distinguish the following two cases.

Case 1. $\Theta(c_K) \in \mathcal{R}$.

Let $f \in \Theta(\mathcal{B}^{(\varphi,\Phi)})$ and let $n \in \mathbb{N}$, and let k be any φ-program for $\Theta(c_K)$. Then, on input f^n the desired machine M outputs k as long as $\Theta(c_K)(x) = f(x)$ for all $x \leq n$. Consequently, if $f = \Theta(c_K)$ then M EX-learns f, and we are done.

Next, assume $f \neq \Theta(c_K)$. Hence, there must be a least $x_0 \in \mathbb{N}$ such that $f(x_0) \neq \Theta(c_K)(x_0)$. As soon as M has found this x_0 by performing the consistency check described above, it behaves as follows. First, it computes $x_1 := \Delta\Theta(c_K)(x_0)$. [6] Next, it calculates $t_n = \max\{\Phi_y(y) \mid y \leq x_1 \wedge \Phi_y(y) \leq$

[5] A mapping $\Xi\colon \mathcal{P} \longrightarrow \mathcal{P}$ is said to be an effective operator if there exists a function $\alpha \in \mathcal{R}$ such that $\Xi(\varphi_i) = \varphi_{\alpha(i)}$ for all $i \in \mathbb{N}$. An effective operator Ξ is total effective if $\Xi(f) \in \mathcal{R}$ for all $f \in \mathcal{R}$ (cf., e.g., [16,30]).

[6] Every general recursive operator Θ can be realized by an algorithmic device $\mathcal{D}_\Theta$: Let g be any total function and let $\Theta(g) = h$. When $\mathcal{D}_\Theta$ is fed the sequence $g(0), g(1), \ldots,$ it does Turing-machine computations and outputs natural numbers. These numbers, in the order of their generation, are defined to be $h(0), h(1), \ldots, .$

$n\}$. Finally, M simulates the learner given in [11] on input f^n and t_n by using the hypothesis space $(\varphi_{\alpha(\tau(i))})_{i\in\mathbb{N}}$, and outputs the resulting hypothesis.

It remains to show that M EX-learns f. This can be seen as follows. First, the sequence $(t_n)_{n\in\mathbb{N}}$ clearly converges to the number $t = \max\{\Phi_y(y) \mid y \le x_1 \ \wedge \ y \in K\}$. Second, suppose any index $i > t$. Since Φ is a coding complexity measure, we know that $\Phi_i(x) \ge i > t$ for all $x \in \mathbb{N}$. Therefore, the definition of $\mathcal{B}^{(\varphi,\Phi)}$ implies $\varphi_{\tau(i)}(x) = c_K(x)$ for all $x \le x_1$. Hence, $\Theta(c_K)(x_0) = \Theta(\varphi_{\tau(i)})(x_0)$, too, and consequently $\Theta(\varphi_{\tau(i)}) \ne f$. But this means that the target function f must be the image of one the remaining functions $\varphi_{\tau(0)}, \ldots, \varphi_{\tau(t)}$. Thus, one of the functions $\varphi_{\alpha(\tau(0))}, \ldots, \varphi_{\alpha(\tau(t))}$ computes f, and consequently Freivalds and Wiehagen's [11] learner correctly identifies f.

Case 2. $\Theta(c_K) \notin \mathcal{R}$.

In this case, the proof is conceptually identical to the demonstration in Case 1. Again let $f \in \Theta(\mathcal{B}^{(\varphi,\Phi)})$ and note that $f \ne \Theta(c_K)$. The major difference consists in incorporating two more limiting effective computations, i.e., one for finding x_0 and one for computing x_1. This can be done as follows. For all $n,\, x \in \mathbb{N}$ let

$$a_n(x) = \begin{cases} 1, & \text{if } \Phi_x(x) \le n, \\ 0, & \text{otherwise.} \end{cases}$$

Then, we set $x_0^n = \min\{y \mid y = n \ \vee \ f(y) \ne \Theta(a_n)(y)\}$ as well as $x_1^n = \Delta\Theta(a_n)(x_0^n)$.

Now, it is easy to see that the sequences $(x_0^n)_{n\in\mathbb{N}}$ and $(x_1^n)_{n\in\mathbb{N}}$ do converge in the limit to numbers x_0 and x_1, respectively, and that x_0 and x_1 have the same properties as their counterparts in the proof of Case 1. We omit the details. ∎

As an inspection of the proof of Theorem 16 shows, we did not use the property $\Phi_i \in \mathcal{R}_{mon}$ at all. Thus, we directly obtain the following corollary.

COROLLARY 17. *For every Gödel numbering φ and every associated coding complexity function $\Phi \in \mathcal{P}^2$ the class $\tilde{\mathcal{B}}^{(\varphi,\Phi)}$ is robustly EX-learnable.*

Moreover, the proof of Theorem 16 can be easily modified to show the robust EX-learnability of the class $\tilde{\mathcal{U}}(K)^{(\varphi,\Phi)}$ for all coding complexity functions.

Then $\Delta\Theta(g)(x)$ is defined to be the least number n such that $\mathcal{D}_\Theta$, when fed $g(0), g(1), \ldots, g(n)$, outputs $h(x)$.

Furthermore, K could be replaced by any r.e. set A, and we have the following corollary, too.

COROLLARY 18. *For every r.e. set A, every Gödel numbering φ and every associated coding complexity function $\Phi \in \mathcal{P}^2$ the class $\tilde{\mathcal{U}}(A)^{(\varphi,\Phi)}$ is robustly EX-learnable.*

Since every class in NUM is robustly EX-learnable, in particular the classes $\mathcal{U}(A)$ for a recursive set A (cf. Theorem 5) are robustly EX-learnable, too.

However, again the situation changes considerably for interpolating complexity measures as our next theorem shows.

THEOREM 19. $\mathcal{B}^{(\varphi,\Phi)} \notin$ robust-EX *for every interpolating complexity measure (φ, Φ).*

Proof. There is a recursive subset $B = \{b_0, b_1, \ldots\}$ of K such that $b_k < b_{k+1}$ and $\Phi_{b_k}(b_k) + 2 \leq \Phi_{b_{k+1}}(b_{k+1})$ for all k. There is a uniform m-reduction h such that $h(k,x) > b_{2(k+x)} \geq k + x$ and $c_{W_k}(x) = c_K(h(k,x))$ for all k, x. Now it is shown that $\Theta(\mathcal{B}^{(\varphi,\Phi)}) = \mathcal{R}_{0,1}$ for the operator Θ defined below and thus $\mathcal{B}^{(\varphi,\Phi)}$ is not robustly BC-learnable if defined by any interpolating measure:

$$\Theta(f)(x) = \begin{cases} f(b_{2x+1}), & \text{if } f(b_{2k}) \geq 1 \text{ for all } k \leq x; \\ f(h(k,x)), & \text{for the first } k \leq x \text{ with } f(b_{2k}) = 0, \text{ otherwise.} \end{cases}$$

The reason for using the condition "$f(b_{2k}) \geq 1$" instead of "$f(b_{2k}) = 1$" is just to make the operator total and giving some value-function $\Theta(f)$ also for the non-$\{0,1\}$-valued functions f.

It is easy to see that the operator Θ is general recursive. Let W_k be a recursive set. Then the function

$$g(y) = \max(\{0\} \cup \{y + \Phi_{h(k,x)}(h(k,x)) \mid x \leq y\})$$

is recursive and satisfies $g(h(k,x)) \geq \Phi_{h(k,x)}(h(k,x))$ for all $x \in W_k$. According to Claim 3 in Theorem 2 there is a complexity function Φ_i such that Φ_i is total, strictly increasing and $\Phi_i(y) \geq g(y)$ for all y. In particular $\Phi_{h(k,x)}(h(k,x)) < \Phi_i(h(k,x))$ for every $x \in W_k$. Φ_i has a finite variant Φ_j with

- $\Phi_j(b_{2x}) = \Phi_{b_{2x}}(b_{2x}) + 1$ and $\Phi_j(b_{2x+1}) = \Phi_{b_{2x}}(b_{2x}) - 1 + 2c_{W_k}(x)$ for $x < k$;
- $\Phi_j(b_{2k}) = \Phi_{b_{2k}}(b_{2k}) - 1$;
- $\Phi_j(y) = \max\{\Phi_{2k}(b_{2k}), \Phi_i(y)\}$ for $y > b_{2k}$;
- Φ_j is monotonically increasing.

Note that the conditions are compatible. Since Φ_i is total, so is Φ_j. Furthermore, by construction and Definition 6, $\varphi_{\tau(j)}(b_{2x}) = 1$ and $\varphi_{\tau(j)}(b_{2x+1}) =$

$c_{W_k}(x)$ for $x < k$ as well as $\varphi_{\tau(j)}(b_{2k}) = 0$. It follows that $\Theta(\varphi_{\tau(j)})(x) = c_{W_k}(x)$ for $x < k$ and that $\Theta(\varphi_{\tau(j)})(x)$ is computed by the second case in the above case-distinction for $x \geq k$. The domination property from Φ_i inherits to Φ_j as the values $h(k,x)$ are beyond b_{2k} for all k,x. Therefore, $h(k,x)$ is enumerated into K iff it is enumerated into K within $\Phi_i(h(k,x))$ steps and $W_k(x) = \varphi_{\tau(j)}(h(k,x))$. It follows that $\Theta(\varphi_{\tau(j)})$ is the characteristic function c_{W_k} of W_k and this completes the proof. ∎

Although the class $\mathcal{B}^{(\varphi,\Phi)}$ is quite natural for all complexity measures (φ,Φ) and its definition does not involve any self-referential coding, the class $\mathcal{B}^{(\varphi,\Phi)}$ is not robustly EX-learnable for all interpolating complexity measures. So while on the one hand the notion of robust EX-learning still permits topological coding tricks [19,29], it does, on the other hand, rule out the natural class $\mathcal{B}^{(\varphi,\Phi)}$ for such natural complexity measures as space. Additionally, the notion of robust EX-learning cannot cope with coding tricks as used in the proof of Theorem 16.

Thus, the examples provided by Jain *et al.* and by our Theorems 16 and 19 give some incidence, that there is still research necessary to find an adequate notion for a "naturally EX-learnable class."

8 Conclusions

The main topic of the present investigations has been the class $\mathcal{B}$ of Blum and Blum [6] and the natural generalizations $\mathcal{U}(A)$ of it obtained by using r.e. sets A as a parameter. It is has been shown that for large families of r.e. sets A, these classes $\mathcal{U}(A)$ are not in NUM. Furthermore, they can be always EX-learned. Moreover, for some but not all sets A there is also a $\mathcal{R}EX$-learner.

Additionally, we studied the learnability properties of the non-monotone counterparts of these classes. While all results obtained for the original classes did not depend on the complexity measures involved, for non-monotone approximation classes of r.e. but non-recursive sets A the situation turned out to be much more subtle. On the one hand, for coding complexity functions there is always an EX-learner. On the other hand, for more natural complexity measures the non-monotone approximation classes are even not BC-learnable.

Furthermore, we dealt with the problem whther or not the class $\mathcal{B}$ is robustly EX-learnable. Again, the answer did depend on the complexity measures involved.

Since $\mathcal{B}$ and the classes $\mathcal{U}(A)$ are quite natural, our results add some incidence that "natural learnability" does not coincide with robust learnability as defined in the current research.

Future work might address the remaining unsolved question whether $\mathcal{U}(A)$ is outside NUM for all non-recursive sets A. Additionally, one might investigate whether $\mathcal{U}(A)$ is robustly BC-learnable for some sets A such that $\mathcal{U}(A)$ is not robustly EX-inferable. It would be also interesting to know whether or not $\mathcal{U}(A)$ can be reliably BC-learned for sets A with $\mathcal{U}(A) \notin \mathcal{R}EX$ (cf. [22] for more information concerning reliable BC-learning). Finally, there are some ways to generalize the notion of $\mathcal{U}(A)$ to every K-recursive set A and one might investigate the learning theoretic properties of the so obtained classes.

Acknowledgement

We would like to thank the referees for many valuable comments and suggestions which have resulted in several improvements of the presentation of the paper.

References

[1] A. Ambainis and R. Freivalds. Transformations that preserve learnability. In *Proccedings of the 7th International Workshop on Algorithmic Learning Theory* (ALT'96) (S. Arikawa and A. Sharma, Eds.) Lecture Notes in Artificial Intelligence Vol. 1160, pages 299–311, Springer-Verlag, Berlin, 1996.

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

[3] D. Angluin and C.H. Smith. A survey of inductive inference: Theory and methods. *Computing Surveys*, 15:237–289, 1983.

[4] J. Bārzdiņš. Prognostication of automata and functions. *Information Processing* 1971, (1) 81–84. Edited by C. P. Freiman, North-Holland, Amsterdam, 1971.

[5] M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the Association for Computing Machinery*, 14:322–336, 1967.

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

[7] J. Case, S. Jain, M. Ott, A. Sharma and F. Stephan. Robust learning aided by context. In *Proceedings of 11th Annual Conference on Computational Learning Theory* (COLT), pages 44–55, ACM Press, New York, 1998.

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

[9] J. Dekker. A theorem on hypersimple sets. *Proceedings of the American Mathematical Society*, 5:791–796, 1954.

[10] R. Freivalds. Inductive inference of recursive functions: Qualitative theory. In *Baltic Computer Science* (J. Bārzdiņš and D. Bjørner, Eds.), Lecture Notes in Computer Science Vol. 502, pages 77–110. Springer-Verlag, Berlin, 1991.

[11] R.V. Freivalds and R. Wiehagen. Inductive inference with additional information. *Elektronische Informationsverarbeitung und Kybernetik*, 15:179–185, 1979.

[12] M. Fulk. Robust separations in inductive inference. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science* (FOCS), pages 405–410, St. Louis, Missouri, 1990.

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

[14] J. Grabowski. Starke Erkennung. *In* Strukturerkennung diskreter kybernetischer Systeme, (R. Linder, H. Thiele, Eds.), Seminarberichte der Sektion Mathematik der Humboldt-Universität Berlin Vol. 82, pages 168–184, 1986.

[15] J. Hartmanis. On the problem of finding natural computational complexity measures. *In* Mathematical Foundations of Computer Science: Proceedings of Symposium and Summer School, Mathematical Institute of the Slovak Academy of Sciences, pp. 95–103, 1973.

[16] J.P. Helm. On effectively computable operators. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* (ZML), 17:231–244, 1971.

[17] S. Jain. Robust behaviorally correct learning. *Information and Computation*, 153:238–248, 1999.

[18] S. Jain, D. Osherson, J.S. Royer and A. Sharma. *Systems That Learn: An Introduction to Learning Theory.* MIT-Press, Boston, MA., 1999.

[19] S. Jain, C. Smith and R. Wiehagen. Robust learning is rich. *Journal of Computer and System Sciences*, to appear.

[20] K.P. Jantke and H.R. Beick. Combining postulates of naturalness in inductive inference. *Journal of Information Processing and Cybernetics* (EIK), 17:465–484, 1981.

[21] E.B. Kinber and T. Zeugmann. Inductive inference of almost everywhere correct programs by reliably working strategies. *Journal of Information Processing and Cybernetics* (EIK), 21:91–100, 1985.

[22] E.B. Kinber and T. Zeugmann. One-sided error probabilistic inductive inference and reliable frequency identification. *Information and Computation*, 92:253–284, 1991.

[23] R. Klette and R. Wiehagen. Research in the theory of inductive inference by GDR mathematicians – A survey. *Information Sciences*, 22:149–169, 1980.

[24] S. Kurtz and C.H. Smith. On the role of search for learning. In *Proceedings of the 2nd Annual Workshop on Computational Learning Theory* (COLT) (R. Rivest, D. Haussler and M. Warmuth, Eds.) pages 303–311, Morgan Kaufman, 1989.

[25] G. Lischke. Über die Erfüllung gewisser Erhaltungssätze durch Kompliziertheitsmaße. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* (ZML), 21:159–166, 1975.

[26] G. Lischke. Flußbildmaße – Ein Versuch zur Definition natürlicher Kompliziertheitsmaße. *Elektronische Informationsverarbeitung und Kybernetik*, 11:423–436, 1975.

[27] E. Minicozzi. Some natural properties of strong-identification in inductive inference. *Theoretical Computer Science*, 2:345–360, 1976.

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

[29] M. Ott and F. Stephan. Avoiding coding tricks by hyperrobust learning. *Proceedings of the Fourth European Conference on Computational Learning Theory* (EuroCOLT) (P. Fischer and H.U. Simon, Eds.) Lecture Notes in Artificial Intelligence Vol. 1572, pages 183–197, Springer-Verlag, Berlin, 1999.

[30] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw–Hill, New York, 1967.

[31] R. Wiehagen. A thesis in inductive inference. In *Proceedings of the 1st International Workshop on Nonmonotonic and Inductive Logic* (NIL'90), (J. Dix, K.P. Jantke, P.H. Schmitt, Eds.), Lecture Notes in Artificial Intelligence Vol. 543, pages 184–207. Springer-Verlag, Berlin, 1991.

[32] T. Zeugmann. A-posteriori characterizations in inductive inference of recursive functions. *Journal of Information Processing and Cybernetics* (EIK), 19:559–594, 1983.

[33] T. Zeugmann. On the nonboundability of total effective operators. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* (ZML), 30:169–172, 1984.

[34] T. Zeugmann. On Bārzdiņš' conjecture. In *Proceedings of the International Workshop on Analogical and Inductive Inference* (AII'86) (K.P. Jantke, Ed.), Lecture Notes in Computer Science Vol. 265, pages 220–227. Springer-Verlag, Berlin, 1986.