Learning Recursive Functions Refutably *

Sanjay Jain[1,**], Efim Kinber[2], Rolf Wiehagen[3], and Thomas Zeugmann[4]

[1] School of Computing, National University of Singapore, Singapore 119260
sanjay@comp.nus.edu.sg
[2] Department of Computer Science, Sacred Heart University, Fairfield, CT 06432-1000, U.S.A.
kinbere@sacredheart.edu
[3] Department of Computer Science, University of Kaiserslautern, PO Box 3049, 67653 Kaiserslautern, Germany
wiehagen@informatik.uni-kl.de
[4] Institut für Theoretische Informatik, Med. Universität zu Lübeck, Wallstraße 40, 23560 Lübeck, Germany
thomas@tcs.mu-luebeck.de

Abstract. Learning of recursive functions refutably means that for *every* recursive function, the learning machine has either to learn this function or to refute it, i.e., to signal that it is not able to learn it. Three modi of making precise the notion of refuting are considered. We show that the corresponding types of learning refutably are of strictly increasing power, where already the most stringent of them turns out to be of remarkable topological and algorithmical richness. All these types are closed under union, though in different strengths. Also, these types are shown to be different with respect to their intrinsic complexity; two of them do not contain function classes that are "most difficult" to learn, while the third one does. Moreover, we present characterizations for these types of learning refutably. Some of these characterizations make clear where the refuting ability of the corresponding learning machines comes from and how it can be realized, in general.
For learning with anomalies refutably, we show that several results from standard learning without refutation stand refutably. Then we derive hierarchies for refutable learning. Finally, we show that stricter refutability constraints cannot be traded for more liberal learning criteria.

1. Introduction

The basic scenario in learning theory informally consists in that a learning machine has to learn some unknown object based on certain information, that is the machine creates one or more hypotheses which eventually converge to a more or less correct and complete description of the object. In learning *refutably* the main goal is more involved. Here, for *every* object from a given universe, the

* A full version of this paper is available as technical report (cf. [17]).
** Supported in part by NUS grant number RP3992710.

learning machine has either to learn the object or to refute it, that is to "signal" if it is incapable to learn this object. This approach is philosophically motivated by Popper's logic of scientific discovery, (testability, falsifyability, refutability of scientific hypotheses), see [31, 24]. Moreover, this approach has also some rather practical implications. If the learning machine signals its inability to learn a certain object, then one can react upon this inability, by modifying the machine, by changing the hypothesis space, or by weakening the learning requirements.

A crucial point of learning refutably is to formally define how the machine is allowed or required to refute a non-learnable object. Mukouchi and Arikawa [29], required refuting to be done in a "one shot" manner, i.e., if after some finite amount of time, the machine concludes that it cannot learn the target object, then it outputs a special "refuting symbol" and stops the learning process forever. Two weaker possibilities of refuting are based on the following observation. Suppose that at some time, the machine feels unable to learn the target object and outputs the refuting symbol. Nevertheless, this time the machine keeps trying to learn the target. It may happen that the information it further receives contains new evidence causing it to change its mind about its inability to learn the object. This process of "alternations" can repeat. It may end in learning the object. Or it may end in refuting it by never revising the machine's belief that it cannot learn the object, i.e., by forever outputting the refuting symbol from some point on. Finally, there may be infinitely many such alternations between trying to learn and believing that this is impossible. In our paper, we will allow and study all three of these modes of learning refutably.

Our universe is the class $\mathcal{R}$ of all recursive functions. The basic learning criterion used is **Ex**, learning in the limit (cf. Definition 1). We study the following types of learning refutably:

RefEx, where refuting a non-learnable function takes place in the *one shot* manner described above (cf. Definition 5).

WRefEx, where both learning and refuting are *limiting* processes, that is on every function from the universe, the learning machine converges either to a correct hypothesis for this function or to the refuting symbol, see Definition 6, (**W** stands for "weak").

RelEx, where a function is considered to be refuted if the learner outputs the refuting symbol *infinitely often* on this function (cf. Definition 7). **Rel** stands for "reliable", since **RelEx** coincides with reliable learning (cf. Proposition 1).

Note that for all types of learning refutably, every function from $\mathcal{R}$ is either learned or refuted by every machine learning refutably. So, it can *not* happen that such a machine converges to an *incorrect* hypothesis (cf. Correctness Lemma).

We show that the types of learning refutably are of strictly increasing power (cf. Theorem 3). Already the most stringent of them, **RefEx**, is of remarkable topological and algorithmical richness (cf. Proposition 3 and Corollary 9). All of these learning types are closed under union, Proposition 5, where **RefEx** and **WRefEx**, on the one hand, and **RelEx**, on the other hand, do not behave completely analogous. Such a difference can also be exhibited with respect to

the intrinsic complexity; actually, both **RefEx** and **WRefEx** do not contain function classes that are "most difficult" to learn, while **RelEx** does contain such classes (cf. Theorems 6 and 7). We also present characterizations for our types of learning refutably. Some of these characterizations make it clear where the refuting ability of the corresponding learning machines comes from and how it can be realized, in general (cf. Theorems 12 and 13).

Besides pure **Ex**-learning refutably we also consider **Ex**-learning and **Bc**-learning with *anomalies* refutably (cf. Definitions 18 and 19). We show that many results from learning without refutation stand refutably, see Theorems 15 and 21. Then we derive several hierarchies for refutable learning, thereby solving an open problem from [22], see Corollaries 16 and 22. Finally, we show that, in general, one cannot trade a stricter refutability constraint for a more liberal learning criterion (cf. Corollary 25 and Theorem 26).

Since the pioneering paper [29] learning with refutation has attracted much attention (cf. [30, 24, 16, 28, 19, 15]).

2. Notation and Preliminaries

Unspecified notations follow [33]. $\mathbb{N}$ denotes the set of natural numbers. We write $\emptyset$ for the empty set and $\mathrm{card}(S)$ for the cardinality of the set S. The minimum and maximum of a set S are denoted by $\min(S)$ and $\max(S)$, respectively.

η, with or without decorations ranges over partial functions. If η_1 and η_2 are both undefined on input x, then, we take $\eta_1(x) = \eta_2(x)$. We say that $\eta_1 \subseteq \eta_2$ iff for all x in the domain of η_1, $\eta_1(x) = \eta_2(x)$. We let $\mathrm{dom}(\eta)$ and $\mathrm{rng}(\eta)$, respectively, denote the domain and range of the partial function η. $\eta(x)\downarrow$ and $\eta(x) = \downarrow$ both denote that $\eta(x)$ is defined and $\eta(x)\uparrow$ as well as $\eta(x) = \uparrow$ stand for $\eta(x)$ is undefined. For any partial functions η, η' and $a \in \mathbb{N}$, we write $\eta =^a \eta'$ and $\eta =^* \eta'$ iff $\mathrm{card}(\{x \mid \eta(x) \neq \eta'(x)\}) \leq a$ and $\mathrm{card}(\{x \mid \eta(x) \neq \eta'(x)\}) < \infty$, respectively. We identify a partial function η with its graph $\{(x, \eta(x)) \mid x \in \mathrm{dom}(\eta)\}$.

For $r \in \mathbb{N}$, the r-extension of η denotes the function f defined as $f(x) = \eta(x)$, if $x \in \mathrm{dom}(\eta)$ and $f(x) = r$, otherwise.

$\mathcal{R}$ denotes the class of all *recursive* functions over $\mathbb{N}$. Furthermore, we set $\mathcal{R}_{0,1} = \{f \mid f \in \mathcal{R} \ \&\ \mathrm{rng}(f) \subseteq \{0,1\}\}$. $\mathcal{C}$ and $\mathcal{S}$, with or without decorations range over subsets of $\mathcal{R}$. For $\mathcal{C} \subseteq \mathcal{R}$, we let $\overline{\mathcal{C}}$ denote $\mathcal{R} \setminus \mathcal{C}$. By $\mathcal{P}$ we denote the class of all *partial recursive* functions over $\mathbb{N}$. f, g, h and F, with or without decorations range over recursive functions unless otherwise specified.

A computable numbering (or just numbering) is a partial recursive function of two arguments. For a numbering $\psi(\cdot,\cdot)$, we use ψ_i to denote the function $\lambda x.\psi(i,x)$, i.e., ψ_i is the function computed by the program i in the numbering ψ. ψ and ϱ range over numberings. $\mathcal{P}_\psi$ denotes the set of partial recursive functions in the numbering ψ, i.e., $\mathcal{P}_\psi = \{\psi_i \mid i \in \mathbb{N}\}$ and $\mathcal{R}_\psi = \{\psi_i \mid i \in \mathbb{N} \ \&\ \psi_i \in \mathcal{R}\}$. That is, $\mathcal{R}_\psi$ stands for the set of all recursive functions in the numbering ψ. A numbering ψ is called one-to-one iff $\psi_i \neq \psi_j$ for any distinct i, j. By φ we denote

a *fixed* acceptable programming system (cf. [33]). We write φ_i for the partial recursive function computed by program i in the φ-system. By Φ we denote any Blum [6] complexity measure associated with φ. We assume without loss of generality that $\Phi_i(x) \geq x$, for all i, x.

$\mathcal{C} \subseteq \mathcal{R}$ is said to be *recursively enumerable* (abbr. r.e.) iff there is an r.e. set X such that $\mathcal{C} = \{\varphi_i \mid i \in X\}$. For any r.e. class $\mathcal{C} \neq \emptyset$, there is an $f \in \mathcal{R}$ such that $\mathcal{C} = \{\varphi_{f(i)} \mid i \in \mathbb{N}\}$.

A function g is called *accumulation point* of a class $\mathcal{C} \subseteq \mathcal{R}$ iff $g \in \mathcal{R}$ and $(\forall n \in \mathbb{N})(\exists f \in \mathcal{C})[(\forall x \leq n)[g(x) = f(x)] \ \& \ f \neq g]$. Note that g may or may not belong to $\mathcal{C}$. For $\mathcal{C} \subseteq \mathcal{R}$, we let $\mathrm{Acc}(\mathcal{C}) = \{g \mid g \text{ is an accumulation point of } \mathcal{C}\}$.

The quantifier $\forall^\infty$ stands for all but finitely many. The following function and class are considered below. Zero is the everywhere 0 function, and $\mathit{FINSUP} = \{f \mid f \in \mathcal{R} \ \& \ (\forall^\infty x)[f(x) = 0]\}$ is the class of all functions of finite support.

2.1. *Function Learning*

We assume that the graph of a function is fed to a machine in canonical order. For a partial function η with $\eta(x)\downarrow$ for all $x < n$, we write $\eta[n]$ for the set $\{(x, \eta(x)) \mid x < n\}$, the finite initial segment of η of length n. We set $\mathrm{SEG} = \{f[n] \mid f \in \mathcal{R} \ \& \ n \in \mathbb{N}\}$ and $\mathrm{SEG}_{0,1} = \{f[n] \mid f \in \mathcal{R}_{0,1} \ \& \ n \in \mathbb{N}\}$. We let σ, τ and γ, with or without decorations range over SEG. Λ is the empty segment. We assume a computable ordering of the elements of SEG.

Let $|\sigma|$ denote the length of σ. Thus, $|f[n]| = n$, for every total function f and all $n \in \mathbb{N}$. If $|\sigma| \geq n$, then we let $\sigma[n]$ denote $\{(x, \sigma(x)) \mid x < n\}$. An *inductive inference machine* (IIM) $\mathbf{M}$ is an algorithmic device that computes a total mapping from SEG into $\mathbb{N}$ (cf. [13]). We say that $\mathbf{M}(f)$ converges to i (written: $\mathbf{M}(f)\downarrow = i$) iff $(\forall^\infty n)[\mathbf{M}(f[n]) = i]$; $\mathbf{M}(f)$ is undefined if no such i exists. Now, we define several criteria of function learning.

Definition 1 ([13, 5, 10]). Let $a \in \mathbb{N} \cup \{*\}$, let $f \in \mathcal{R}$ and let $\mathbf{M}$ be an IIM.

(a) $\mathbf{M}$ $\mathbf{Ex}^a$*-learns* f (abbr. $f \in \mathbf{Ex}^a(\mathbf{M})$) iff there is an i with $\mathbf{M}(f)\downarrow = i$ and $\varphi_i =^a f$.
(b) $\mathbf{M}$ $\mathbf{Ex}^a$*-learns* $\mathcal{C}$ iff $\mathbf{M}$ $\mathbf{Ex}^a$-learns each $f \in \mathcal{C}$.
(c) $\mathbf{Ex}^a = \{\mathcal{C} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{C} \subseteq \mathbf{Ex}^a(\mathbf{M})]\}$.

Note that for $a = 0$ we omit the upper index, i.e., we set $\mathbf{Ex} = \mathbf{Ex}^0$.

By the definition of convergence, only finitely many data of f were seen by an IIM up to the (unknown) point of convergence. Hence, some learning must have taken place. Thus, we use *identify*, *learn* and *infer* interchangeably.

Definition 2 ([2, 10]). Let $a \in \mathbb{N} \cup \{*\}$, let $f \in \mathcal{R}$ and let $\mathbf{M}$ be an IIM.

(a) $\mathbf{M}$ $\mathbf{Bc}^a$*-learns* f (written: $f \in \mathbf{Bc}^a(\mathbf{M})$) iff $(\forall^\infty n)[\varphi_{\mathbf{M}(f[n])} =^a f]$.
(b) $\mathbf{M}$ $\mathbf{Bc}^a$*-learns* $\mathcal{C}$ iff $\mathbf{M}$ $\mathbf{Bc}^a$-learns each $f \in \mathcal{C}$.
(c) $\mathbf{Bc}^a = \{\mathcal{C} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{C} \subseteq \mathbf{Bc}^a(\mathbf{M})]\}$.

We set $\mathbf{Bc} = \mathbf{Bc}^0$. Harrington [10] showed that $\mathcal{R} \in \mathbf{Bc}^*$. Thus, we shall consider mainly $\mathbf{Bc}^a$ for $a \in \mathbb{N}$ in the following.

Definition 3 (Minicozzi [27], Blum and Blum [5]). Let **M** be an IIM.
(a) **M** is *reliable* iff for all $f \in \mathcal{R}$, $\mathbf{M}(f)\downarrow \Rightarrow$ **M Ex**-identifies f.
(b) **M RelEx**-*infers* $\mathcal{C}$ (written: $\mathcal{C} \subseteq \mathbf{RelEx}(\mathbf{M})$) iff **M** is reliable and **M Ex**-infers $\mathcal{C}$.
(c) $\mathbf{RelEx} = \{\mathcal{C} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathbf{M}\ \mathbf{RelEx}\text{-infers } \mathcal{C}]\}$.

Thus, a machine is reliable if it does not converge on functions it fails to identify. For references on reliable learning besides [27, 5], see [21, 14, 22, 8].

Definition 4. $\mathbf{NUM} = \{\mathcal{C} \mid (\exists \mathcal{C}' \mid \mathcal{C} \subseteq \mathcal{C}' \subseteq \mathcal{R})[\mathcal{C}' \text{ is recursively enumerable}]\}$.

Inductive inference within **NUM** has been studied, e.g. in [13, 3]. For the general theory of learning recursive functions, see [1, 5, 10, 11, 23, 18].

2.2. *Learning Refutably*

Next, we introduce learning with refutation. We consider three versions of refutation based on how the machine is required to refute a function. First we extend the definition of IIM by allowing it to output a special symbol $\perp$. Thus, now an IIM maps SEG to $\mathbb{N} \cup \{\perp\}$. Convergence of an IIM on a function is defined as before (but now a machine may converge to a number $i \in \mathbb{N}$ or to $\perp$).

Definition 5. Let **M** be an IIM. **M RefEx**-*identifies* a class $\mathcal{C}$ (written: $\mathcal{C} \subseteq \mathbf{RefEx}(\mathbf{M})$) iff the following conditions are satisfied.
(a) $\mathcal{C} \subseteq \mathbf{Ex}(\mathbf{M})$.
(b) For all $f \in \mathbf{Ex}(\mathbf{M})$, for all n, $\mathbf{M}(f[n]) \neq \perp$.
(c) For all $f \in \mathcal{R}$ such that $f \notin \mathbf{Ex}(\mathbf{M})$, there exists an $n \in \mathbb{N}$ such that $(\forall m < n)[\mathbf{M}(f[m]) \neq \perp]$ and $(\forall m \geq n)[\mathbf{M}(f[m]) = \perp]$.

The following generalization of **RefEx** places less restrictive constraint on how the machine refutes a function. **WRef** below stands for weak refutation.

Definition 6. Let **M** be an IIM. **M WRefEx**-*learns* a class $\mathcal{C}$ (written: $\mathcal{C} \subseteq \mathbf{WRefEx}(\mathbf{M})$) iff the following conditions are satisfied.
(a) $\mathcal{C} \subseteq \mathbf{Ex}(\mathbf{M})$.
(b) For all $f \in \mathcal{R}$ such that $f \notin \mathbf{Ex}(\mathbf{M})$, $\mathbf{M}(f)\downarrow = \perp$.

For weakly refuting a function f, an IIM just needs to converge to $\perp$. Before convergence, it may change its mind finitely often whether or not to refute f. Another way an IIM may refute a function f is to output $\perp$ on f infinitely often.

Definition 7. Let **M** be an IIM. **M RelEx**$'$-*identifies* a class $\mathcal{C}$ (written: $\mathcal{C} \subseteq \mathbf{RelEx}'(\mathbf{M})$) iff the following conditions are satisfied.
(a) $\mathcal{C} \subseteq \mathbf{Ex}(\mathbf{M})$.
(b) For all $f \in \mathcal{R}$ such that $f \notin \mathbf{Ex}(\mathbf{M})$, there exists infinitely many $n \in \mathbb{N}$ such that $\mathbf{M}(f[n]) = \perp$.

Proposition 1. $\mathbf{RelEx} = \mathbf{RelEx}'$.

As it follows from their definitions, for any of the learning types **RefEx**, **WRefEx** and **RelEx**, we get that any $f \in \mathcal{R}$ has either to be learned or to be refuted. This is made formally precise by the following Correctness Lemma.

Lemma 1 (*Correctness Lemma*). *Let* $\mathbf{I} \in \{\mathbf{RefEx}, \mathbf{WRefEx}, \mathbf{RelEx}\}$. *For any* $\mathcal{C} \subseteq \mathcal{R}$, *any IIM* $\mathbf{M}$ *with* $\mathcal{C} \subseteq \mathbf{I}(\mathbf{M})$, *and any* $f \in \mathcal{R}$, *if* $\mathbf{M}(f)\downarrow \in \mathbb{N}$, *then* $\varphi_{\mathbf{M}(f)} = f$.

3. Ex-Learning Refutably

We first derive several properties of the defined types of learning refutably. We then relate these types by their so-called intrinsic complexity. Finally, we present several characterizations for refutable learnability.

3.1. *Properties and Relations*

First, we exhibit some properties of refutably learnable classes. These properties imply that the corresponding learning types are of strictly increasing power. Already the most stringent of these types, **RefEx**, is of surprising richness. In particular, every class from **RefEx** can be enriched by including all of its accumulation points. This is not possible for the classes from **WRefEx** and **RelEx**, as it follows from the proof of Theorem 3.

Proposition 2. *For all* $\mathcal{C} \in \mathbf{RefEx}$, $\mathcal{C} \cup \mathrm{Acc}(\mathcal{C}) \in \mathbf{RefEx}$.

Proof. Suppose $\mathcal{C} \in \mathbf{RefEx}$ as witnessed by some total IIM $\mathbf{M}$. Let $g \in \mathcal{R}$ be an accumulation point of $\mathcal{C}$. We claim that $\mathbf{M}$ must **Ex**-identify g. Assume to the contrary that for some n, $\mathbf{M}(g[n]) = \perp$. Then, by the definition of accumulation point, there is a function $f \in \mathcal{C}$ such that $g[n] \subseteq f$. Hence $\mathbf{M}(f[n]) = \perp$, too, a contradiction to $\mathbf{M}$ **RefEx**-identifying $\mathcal{C}$. ∎

The next proposition shows that **RefEx** contains "topologically rich", namely *non-discrete* classes, i.e. classes which contain accumulation points. Thus, **RefEx** is "richer" than **Ex**-learning without any mind change, since any class being learnable in that latter sense may not contain any of its accumulation points (cf. [25]). More precisely, **RefEx** and **Ex**-learning without mind changes are set-theoretically *incomparable*; the missing direction follows from Theorem 14 below.

Proposition 3. RefEx *contains non-discrete classes.*

The following proposition establishes some bound on the topological richness of the classes from **WRefEx**.

Definition 8. A class $\mathcal{C} \subseteq \mathcal{R}$ is called *initially complete* iff for every $\sigma \in \mathrm{SEG}$, there is a function $f \in \mathcal{C}$ such that $\sigma \subseteq f$.

Proposition 4. WRefEx *does not contain any initially complete class.*

The following result is needed for proving Theorem 3 below.

Lemma 2. $\mathcal{C} = \{f \in \mathcal{R} \mid (\forall x \in \mathbb{N})[f(x) \neq 0]\} \notin \mathbf{Ex}$.

We are now ready to prove that **RefEx**, **WRefEx** and **RelEx**, respectively, are of strictly increasing power.

Theorem 3. $\mathbf{RefEx} \subset \mathbf{WRefEx} \subset \mathbf{RelEx}$.

Proof. $\mathbf{RefEx} \subseteq \mathbf{WRefEx} \subseteq \mathbf{RelEx}$ by their definitions and Proposition 1.

We first show that $\mathbf{WRefEx} \setminus \mathbf{RefEx} \neq \emptyset$. For that purpose, we define $\mathrm{SEG}^+ = \{f[n] \mid f \in \mathcal{R} \;\&\; n \in \mathbb{N} \;\&\; (\forall x \in \mathbb{N})[f(x) \neq 0]\}$. Let $\mathcal{C} = \{0\text{-ext}(\sigma) \mid \sigma \in \mathrm{SEG}^+\}$. Then $\mathrm{Acc}(\mathcal{C}) = \{f \in \mathcal{R} \mid (\forall x \in \mathbb{N})[f(x) \neq 0]\}$, which is not in **Ex**, by Lemma 2. Thus, $\mathcal{C} \cup \mathrm{Acc}(\mathcal{C}) \notin \mathbf{Ex}$, and hence, $\mathcal{C} \notin \mathbf{RefEx}$, by Proposition 2.

In order to show that $\mathcal{C} \in \mathbf{WRefEx}$, let $\mathrm{prog} \in \mathcal{R}$ be a recursive function such that for any $\sigma \in \mathrm{SEG}^+$, $\mathrm{prog}(\sigma)$ is a φ-program for $0\text{-ext}(\sigma)$. Let $\mathbf{M}$ be defined as follows.

$$\mathbf{M}(f[n]) = \begin{cases} \perp, & \text{if } f[n] \in \mathrm{SEG}^+; \\ \mathrm{prog}(\sigma), & \text{if } 0\text{-ext}(f[n]) = 0\text{-ext}(\sigma), \text{ for some } \sigma \in \mathrm{SEG}^+; \\ \perp, & \text{otherwise.} \end{cases}$$

It is easy to verify that $\mathbf{M}$ **WRefEx**-identifies $\mathcal{C}$.

We now show that $\mathbf{RelEx} \setminus \mathbf{WRefEx} \neq \emptyset$. *FINSUP* is initially complete and $\mathit{FINSUP} \in \mathbf{NUM}$. Since $\mathbf{NUM} \subseteq \mathbf{RelEx}$, see [27], we have that $\mathit{FINSUP} \in \mathbf{RelEx}$. On the other hand, $\mathit{FINSUP} \notin \mathbf{WRefEx}$ by Proposition 4. ∎

As a consequence from the proof of Theorem 3, we can derive that the types **RefEx**, **WRefEx** and **RelEx** already differ on *recursively enumerable* classes.

Corollary 4. $\mathbf{RefEx} \cap \mathbf{NUM} \subset \mathbf{WRefEx} \cap \mathbf{NUM} \subset \mathbf{RelEx} \cap \mathbf{NUM}$.

We next point out that all the types of learning refutably share a pretty rare, but desirable property, namely to be closed under union.

Proposition 5. $\mathbf{RefEx}, \mathbf{WRefEx}$ *and* $\mathbf{RelEx}$ *are closed under finite union.*

RelEx is even closed under the union of any effectively given *infinite* sequence of classes (cf. [27]). The latter is not true for both **RefEx** and **WRefEx**, as it can be seen by shattering the class *FINSUP* into its subclasses of *one* element each.

3.2. *Intrinsic Complexity*

There is another field where **RefEx** and **WRefEx**, on the one hand, and **RelEx**, on the other hand, behave differently, namely that of intrinsic complexity. The intrinsic complexity compares the difficulty of learning by using some reducibility notion, see [12]. With every reducibility notion comes a notion of completeness. A function class is complete for some learning type I, if this class is "most difficult" to learn among all the classes from I. As we show, **RefEx** and **WRefEx** do not contain such complete classes, while **RelEx** does.

Definition 9. A sequence $P = p_0, p_1, \ldots$ of natural numbers is called **Ex**-*admissible* for $f \in \mathcal{R}$ iff P converges to a program p for f.

Definition 10 (Rogers [33]). A *recursive operator* is an effective total mapping, Θ, from (possibly partial) functions to (possibly partial) functions such that:

(a) For all functions η, η', if $\eta \subseteq \eta'$ then $\Theta(\eta) \subseteq \Theta(\eta')$.
(b) For all η, if $(x, y) \in \Theta(\eta)$, then there is a finite function $\alpha \subseteq \eta$ such that $(x, y) \in \Theta(\alpha)$.
(c) For all finite functions α, one can effectively enumerate (in α) all $(x, y) \in \Theta(\alpha)$.

For each recursive operator Θ, we can effectively find a recursive operator Θ' such that

(d) for each finite function α, $\Theta'(\alpha)$ is finite, and its canonical index can be effectively determined from α, and
(e) for all total functions f, $\Theta'(f) = \Theta(f)$.

This allows us to get a nice effective sequence of recursive operators.

Proposition 6. *There exists an effective enumeration, $\Theta_0, \Theta_1, \cdots$ of recursive operators satisfying condition* (d) *above such that, for all recursive operators Θ, there exists an $i \in \mathbb{N}$ satisfying $\Theta(f) = \Theta_i(f)$ for all total functions f.*

Definition 11 (Freivalds *et al.* [12]). Let $\mathcal{S}, \mathcal{C} \in \mathbf{Ex}$. Then $\mathcal{S}$ is called **Ex**-*reducible* to $\mathcal{C}$ (written: $\mathcal{S} \leq_{\mathbf{Ex}} \mathcal{C}$) iff there exist two recursive operators Θ and Ξ such that for all $f \in \mathcal{S}$,

(a) $\Theta(f) \in \mathcal{C}$,
(b) for any **Ex**-admissible sequence P for $\Theta(f)$, $\Xi(P)$ is **Ex**-admissible for f.

If $\mathcal{S}$ is **Ex**-reducible to $\mathcal{C}$, then $\mathcal{C}$ is at least as difficult to **Ex**-learn as $\mathcal{S}$ is. Indeed, if **M** **Ex**-learns $\mathcal{C}$, then $\mathcal{S}$ is **Ex**-learnable by an IIM that, on any function $f \in \mathcal{S}$, outputs $\Xi(\mathbf{M}(\Theta(f)))$.

Definition 12. Let **I** be a learning type and $\mathcal{C} \subseteq \mathcal{R}$. $\mathcal{C}$ is called **Ex**-*complete* in **I** iff $\mathcal{C} \in \mathbf{I}$, and for all $\mathcal{S} \in \mathbf{I}$, $\mathcal{S} \leq_{\mathbf{Ex}} \mathcal{C}$.

Theorem 5. *Let $\mathcal{C} \in$ **WRefEx**. Then there exists a class $\mathcal{S} \in$ **RefEx** such that $\mathcal{S} \not\leq_{\mathbf{Ex}} \mathcal{C}$.*

Theorem 5 immediately yields the following result.

Theorem 6. (1) *There is no* **Ex**-*complete class in* **RefEx**.
(2) *There is no* **Ex**-*complete class in* **WRefEx**.

In contrast to Theorem 6, **RelEx** contains an **Ex**-complete class.

Theorem 7. *There is an* **Ex**-*complete class in* **RelEx**.

3.3. *Characterizations*

We present several characterizations for **RefEx**, **WRefEx** and **RelEx**. The first group of characterizations relates refutable learning to the established concept of *classification*. The main goal in recursion theoretic classification can be described as follows. Let be given some finite (or even infinite) family of function classes. Then, for an arbitrary function from the union of all these classes, one

has to find out which of these classes the corresponding function belongs to, see [4, 37, 35, 34, 9]. What we need in our characterization theorems below will be classification where only *two* classes are involved in the classification process, more exactly, a class together with its complement; and semi-classification which is some weakening of classification. Note that the corresponding characterizations using these kinds of classification are in a sense close to the definitions of learning refutably. Nevertheless, these characterizations are useful in that their characteristic conditions are easily testable, i.e. they allow to check, whether or not a given class is learnable with refutation.

Let $\mathcal{R}_{0,?}$ be the class of all total computable functions mapping $\mathbb{N}$ into $\{0, ?\}$.

Definition 13. $\mathcal{S} \subseteq \mathcal{R}$ is *finitely semi-classifiable* iff there is $c \in \mathcal{R}_{0,?}$ such that

(a) for every $f \in \mathcal{S}$, there is an $n \in \mathbb{N}$ such that $c(f[n]) = 0$,
(b) for every $f \in \overline{\mathcal{S}}$ and for all $n \in \mathbb{N}$, $c(f[n]) = ?$.

Intuitively, a class $\mathcal{S} \subseteq \mathcal{R}$ is finitely semi-classifiable if for every $f \in \mathcal{S}$ after some finite amount of time one finds out that $f \in \mathcal{S}$, whereas for every $f \in \overline{\mathcal{S}}$, one finds out "nothing".

Theorem 8. *For any $\mathcal{C} \subseteq \mathcal{R}$, $\mathcal{C} \in$ **RefEx** iff $\mathcal{C}$ is contained in some class $\mathcal{S} \in$ **Ex** such that $\overline{\mathcal{S}}$ is finitely semi-classifiable.*

Proof. Necessity. Suppose $\mathcal{C} \in$ **RefEx** as witnessed by some total IIM $\mathbf{M}$. Let $\mathcal{S} = \mathbf{Ex}(\mathbf{M})$. Clearly, $\mathcal{C} \subseteq \mathcal{S}$. Furthermore, (i) for any $f \in \mathcal{S}$ and any $n \in \mathbb{N}$, $\mathbf{M}(f[n]) \neq \perp$, and (ii) for any $f \in \overline{\mathcal{S}}$, there is $n \in \mathbb{N}$ such that $\mathbf{M}(f[n]) = \perp$.

Now define c as follows.

$$c(f[n]) = \begin{cases} 0, & \text{if } \mathbf{M}(f[n]) = \perp; \\ ?, & \text{if } \mathbf{M}(f[n]) \neq \perp. \end{cases}$$

Clearly, $c \in \mathcal{R}_{0,?}$ and $\overline{\mathcal{S}}$ is finitely semi-classifiable by c.

Sufficiency. Suppose $\mathcal{C} \subseteq \mathcal{S} \subseteq \mathbf{Ex}(\mathbf{M})$, and $\overline{\mathcal{S}}$ is finitely semi-classifiable by some $c \in \mathcal{R}_{0,?}$. Now define $\mathbf{M}'$ as follows.

$$\mathbf{M}'(f[n]) = \begin{cases} \mathbf{M}(f[n]), & \text{if } c(f[n]) = ?; \\ \perp, & \text{if } c(f[x]) = 0, \text{ for some } x \leq n. \end{cases}$$

It is easy to verify that $\mathbf{M}'$ **RefEx**-identifies $\mathcal{C}$. ∎

We can apply the characterization of **RefEx** above in order to show that **RefEx** contains "non-trivial" classes. Therefore, let

$$\mathcal{C} = \{f \mid f \in \mathcal{R} \;\&\; \varphi_{f(0)} = f \;\&\; (\forall x \in \mathbb{N})[\Phi_{f(0)}(x) \leq f(x+1)]\}.$$

Clearly, $\mathcal{C} \in$ **Ex** and $\overline{\mathcal{C}}$ is finitely semi-classifiable. Hence, by Theorem 8, $\mathcal{C}$ is **RefEx**-learnable. $\mathcal{C} \notin$ **NUM** was shown in [38], Theorem 4.2. Hence, we get the following corollary illustrating that **RefEx** contains "algorithmically rich" classes, that is classes being not contained in any recursively enumerable class.

Corollary 9. $\mathbf{RefEx} \setminus \mathbf{NUM} \neq \emptyset$.

We now characterize **WRefEx**. Therefore, we need the special case of classification where the classes under consideration form a partition of $\mathcal{R}$.

Definition 14 ([37]). (1) Let $\mathcal{C}, \mathcal{S} \subseteq \mathcal{R}$, where $\mathcal{C} \cap \mathcal{S} = \emptyset$. $(\mathcal{C}, \mathcal{S})$ is called *classifiable* iff there is $c \in \mathcal{R}_{0,1}$ such that for any $f \in \mathcal{C}$ and for almost all $n \in \mathbb{N}$, $c(f[n]) = 0$; and for any $f \in \mathcal{S}$ and for almost all $n \in \mathbb{N}$, $c(f[n]) = 1$.

(2) A class $\mathcal{C} \subseteq \mathcal{R}$ is called *classifiable* iff $(\mathcal{C}, \overline{\mathcal{C}})$ is classifiable.

Theorem 10. *For any $\mathcal{C} \subseteq \mathcal{R}$, $\mathcal{C} \in \mathbf{WRefEx}$ iff $\mathcal{C} \subseteq \mathcal{S}$ for a classifiable class $\mathcal{S} \in \mathbf{Ex}$.*

Proof. Necessity. Suppose $\mathcal{C} \in \mathbf{WRefEx}$ as witnessed by some total IIM $\mathbf{M}$. Let $\mathcal{S} = \mathbf{Ex}(\mathbf{M})$. Clearly, $\mathcal{C} \subseteq \mathcal{S}$ and $\mathcal{S} \in \mathbf{Ex}$. Now define c as follows.

$$c(f[n]) = \begin{cases} 0, & \text{if } \mathbf{M}(f[n]) \neq \perp; \\ 1, & \text{if } \mathbf{M}(f[n]) = \perp. \end{cases}$$

Then, clearly, $\mathcal{S}$ is classifiable by c.

Sufficiency. Suppose $\mathcal{C} \subseteq \mathcal{S} \subseteq \mathbf{Ex}(\mathbf{M})$, and let $\mathcal{S}$ be classifiable by some $c \in \mathcal{R}_{0,1}$. Then, define $\mathbf{M}'$ as follows.

$$\mathbf{M}'(f[n]) = \begin{cases} \mathbf{M}(f[n]), & \text{if } c(f[n]) = 0 \\ \perp, & \text{if } c(f[n]) = 1. \end{cases}$$

Clearly, $\mathbf{M}'$ witnesses that $\mathcal{C} \in \mathbf{WRefEx}$. ∎

Finally, we give a characterization of **RelEx** in terms of semi-classifiability.

Definition 15 ([35]). $\mathcal{S} \subseteq \mathcal{R}$ is *semi-classifiable* iff there is $c \in \mathcal{R}_{0,?}$ such that

(a) for any $f \in \mathcal{S}$ and almost all $n \in \mathbb{N}$, $c(f[n]) = 0$,
(b) for any $f \in \overline{\mathcal{S}}$ and infinitely many $n \in \mathbb{N}$, $c(f[n]) = ?$.

Thus, a class $\mathcal{S}$ of recursive functions is semi-classifiable if for every function $f \in \mathcal{S}$, one can find out in the limit that f belongs to $\mathcal{S}$, while for any $g \in \mathcal{R} \setminus \mathcal{S}$ one is not required to know in the limit where this function g comes from.

Theorem 11. *For all $\mathcal{C} \subseteq \mathcal{R}$, $\mathcal{C} \in \mathbf{RelEx}$ iff $\mathcal{C} \subseteq \mathcal{S}$ for a semi-classifiable class $\mathcal{S} \in \mathbf{Ex}$.*

Proof. Necessity. Suppose $\mathcal{C} \in \mathbf{RelEx}$ by some total IIM $\mathbf{M}$. Let $\mathcal{S} = \mathbf{Ex}(\mathbf{M})$. Clearly, $\mathcal{C} \subseteq \mathcal{S}$. In order to show that $\mathcal{S}$ is semi-classifiable, define c as follows.

$$c(f[n]) = \begin{cases} 0, & \text{if } n = 0 \text{ or } \mathbf{M}(f[n-1]) = \mathbf{M}(f[n]); \\ ?, & \text{if } n > 0 \text{ and } \mathbf{M}(f[n-1]) \neq \mathbf{M}(f[n]). \end{cases}$$

Now, for any $f \in \mathcal{S}$, $\mathbf{M}(f)\downarrow$, and thus $c(f[n]) = 0$ for almost all $n \in \mathbb{N}$. On the other hand, if $f \in \overline{\mathcal{S}}$ then $f \notin \mathbf{Ex}(\mathbf{M})$. Consequently, since $\mathbf{M}$ is reliable and total, we have $\mathbf{M}(f[n-1]) \neq \mathbf{M}(f[n])$ for infinitely many $n \in \mathbb{N}$. Hence $c(f[n]) = ?$ for infinitely many n. Thus, $\mathcal{S}$ is semi-classifiable by c.

Sufficiency. Suppose $\mathcal{C} \subseteq \mathcal{S} \subseteq \mathbf{Ex}(\mathbf{M})$. Suppose $\mathcal{S}$ be semi-classifiable by some $c \in \mathcal{R}_{0,?}$. Define $\mathbf{M}'$ as follows.

$$\mathbf{M}'(f[n]) = \begin{cases} \mathbf{M}(f[n]), & \text{if } c(f[n]) = 0; \\ n, & \text{if } c(f[n]) = ?. \end{cases}$$

Now, for any $f \in \mathcal{S}$, for almost all n, $c(f[n]) = 0$. Hence $\mathbf{M}'$ will $\mathbf{Ex}$-learn f, since $\mathbf{M}$ does so. If $f \in \overline{\mathcal{S}}$, then $c(f[n]) = ?$ for infinitely many n. Consequently, $\mathbf{M}'$ diverges on f caused by arbitrarily large outputs. Thus, $\mathbf{M}'$ $\mathbf{RelEx}$-learns $\mathcal{C}$. ∎

There is a kind of "dualism" in the characterizations of **RefEx** and **RelEx**. A class is **RefEx**-learnable if it is contained in some **Ex**-learnable class having a *complement* that is finitely semi-classifiable. In contrast, a class is **RelEx**-learnable if it is subset of an **Ex**-learnable class that *itself* is semi-classifiable.

The characterizations of the second group, this time for **RefEx** and **RelEx**, significantly differ from the characterizations presented above in two points. First, the characteristic conditions are stated here in terms that formally have nothing to do with learning. Second, the sufficiency proofs are again constructive and they make clear where the "refuting ability" of the corresponding learning machines in general comes from. For stating the corresponding characterization of **RefEx**, we need the following notions.

Definition 16. A numbering ψ is *strongly one-to-one* iff there is a recursive function $d\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for all $i, j \in \mathbb{N}$, $i \neq j$, there is an $x < d(i,j)$ with $\psi_i(x) \neq \psi_j(x)$.

Any strongly one-to-one numbering is one-to-one. Moreover, given any distinct ψ-indices i and j, the functions ψ_i and ψ_j do not only differ, but one can compute a bound on the least argument on which these functions differ.

Definition 17 ([32]). A class $\Pi \subseteq \mathcal{P}$ is called *completely r.e.* iff $\{i \mid \varphi_i \in \Pi\}$ is recursively enumerable.

Now, we can present our next characterization.

Theorem 12. *For any $\mathcal{C} \subseteq \mathcal{R}$, $\mathcal{C} \in \mathbf{RefEx}$ iff there are numberings ψ and ϱ such that*

(1) *ψ is strongly one-to-one and $\mathcal{C} \subseteq \mathcal{P}_\psi$,*
(2) *$\mathcal{P}_\varrho$ is completely r.e. and $\mathcal{R}_\varrho = \overline{\mathcal{R}_\psi}$.*

By the proof of Theorem 12, in **RefEx**-learning the processes of learning and refuting, respectively, can be nicely separated. An IIM can be provided with two spaces, one for learning, ψ, and one for refuting, ϱ. If and when the "search for refutation" in the refutation space has been successful, the learning process can be stopped forever. This search for refutation is based on the fact that the refutation space forms a completely r.e. class $\mathcal{P}_\varrho$ of partial recursive functions. The spaces for learning and refuting are interconnected by the essential property that their recursive kernels, $\mathcal{R}_\psi$ and $\mathcal{R}_\varrho$, disjointly exhaust $\mathcal{R}$. This property guarantees that each recursive function either will be learned or refuted. The

above characterization of **RefEx** is "more granular" than the one of **RefEx** by Theorem 8. The characterization of Theorem 8 requires that one should find out *anyhow* if the given function does not belong to the target class. The characterization of Theorem 12 makes precise *how* this task can be done. Moreover, the **RefEx**-characterization of Theorem 12 is incremental to a characterization of **Ex**, since the existence of a numbering with condition (1) above is necessary and sufficient for **Ex**-learning the class $\mathcal{C}$ (cf. [36]). Finally, the refutation space could be "economized" in the same manner as the learning space by making it one-to-one.

The following characterization of **RelEx** is a slight modification of a result from [20].

Theorem 13. *For any $\mathcal{C} \subseteq \mathcal{R}$, $\mathcal{C} \in$ **RelEx** iff there are a numbering ψ and a function $d \in \mathcal{R}$ such that*

(1) *for any $f \in \mathcal{R}$, if $H_f = \{i \mid f[d(i)] \subseteq \psi_i\}$ is finite, then H_f contains a ψ-index of f,*
(2) *for any $f \in \mathcal{C}$, H_f is finite.*

Theorem 13 instructively clarifies where the ability to learn reliably may come from. Mainly, it comes from the properties of a well-chosen space of hypotheses. In any such space ψ exhibited by Theorem 13, for any function f from the class to be learned, there are only finitely many "candidates" for ψ-indices of f, the set H_f. This *finiteness* of H_f together with the fact that H_f then contains a ψ-index of f, make sure that the amalgamation technique [10] succeeds in learning any such f. Conversely, the *infinity* of this set H_f of candidates automatically ensures that the learning machine as defined in the sufficiency proof of Theorem 13 *diverges* on f. This is achieved by causing the corresponding machine to output arbitrarily large hypotheses on every function $f \in \mathcal{R}$ with H_f being infinite.

4. $\mathbf{Ex}^a$-Learning and $\mathbf{Bc}^a$-Learning Refutably

In this section, we consider **Ex**-learning and **Bc**-learning *with anomalies* refutably. Again, we will derive both strengths and weaknesses of refutable learning. As it turns out, many results of standard learning, i.e. without refutation, stand refutably. This yields several hierarchies for refutable learning. Furthermore, we show that in general one cannot trade the strictness of the refutability constraints for the liberality of the learning criteria.

We can now define $\mathbf{IEx}^a$ and $\mathbf{IBc}^a$ for $\mathbf{I} \in \{\mathbf{Ref}, \mathbf{WRef}, \mathbf{Rel}\}$ analogously to Definitions 5, 6, and 7. We only give the definitions of $\mathbf{RefEx}^a$ and $\mathbf{RelBc}^a$ as examples.

Definition 18. Let $a \in \mathbb{N} \cup \{*\}$ and let $\mathbf{M}$ be an IIM. $\mathbf{M}$ $\mathbf{RefEx}^a$*-learns* $\mathcal{C}$ iff

(a) $\mathcal{C} \subseteq \mathbf{Ex}^a(\mathbf{M})$.
(b) For all $f \in \mathbf{Ex}^a(\mathbf{M})$, for all n, $\mathbf{M}(f[n]) \neq \perp$.
(c) For all $f \in \mathcal{R}$ such that $f \notin \mathbf{Ex}^a(\mathbf{M})$, there exists an $n \in \mathbb{N}$ such that $(\forall m < n)[\mathbf{M}(f[m]) \neq \perp]$ and $(\forall m \geq n)[\mathbf{M}(f[m]) = \perp]$.

Definition 19 ([22]). Let $a \in \mathbb{N} \cup \{*\}$ and let $\mathbf{M}$ be an IIM. $\mathbf{M}$ **RelBc**^{a}*-learns* $\mathcal{C}$ iff

(a) $\mathcal{C} \subseteq \mathbf{Bc}^a(\mathbf{M})$.
(b) For all $f \in \mathcal{R}$ such that $f \notin \mathbf{Bc}^a(\mathbf{M})$, there exist infinitely many $n \in \mathbb{N}$ such that $\mathbf{M}(f[n]) = \perp$.

RelEx^a and **RelBc**^{a} were studied firstly in [21] and [22], respectively.

Our first result points out some weakness of learning refutably. It shows that there are classes which, on the one hand, are easy to learn in the standard sense of **Ex**-learning without any mind change, but, on the other hand, which are not learnable refutably, even if we allow both the most liberal type of learning refutably, namely reliable learning, and the very rich type of **Bc**-learning with an arbitrarily large number of anomalies. For proving this result, we need the following proposition.

Proposition 7. (a) *For any* $a \in \mathbb{N}$ *and any* $\sigma \in \mathrm{SEG}$, $\{f \in \mathcal{R} \mid \sigma \subseteq f\} \notin \mathbf{Bc}^a$.

(b) *For any* $a \in \mathbb{N}$ *and any* $\sigma \in \mathrm{SEG}_{0,1}$, $\{f \in \mathcal{R}_{0,1} \mid \sigma \subseteq f\} \notin \mathbf{Bc}^a$.

Next, recall that **Ex**-learning without mind changes is called *finite learning*. Informally, here the learning machine has "one shot" only to do its learning task. We denote the resulting learning type by **Fin**.

Theorem 14. *For all* $a \in \mathbb{N}$, $\mathbf{Fin} \setminus \mathbf{RelBc}^a \neq \emptyset$.

Next we show that allowing anomalies can help in learning refutably. Indeed, while $\mathbf{Ex}^{a+1} \setminus \mathbf{Ex}^a \neq \emptyset$ was shown in [10], we now strengthen this result to **RefEx**-learning with anomalies.

Theorem 15. *For any* $a \in \mathbb{N}$, $\mathbf{RefEx}^{a+1} \setminus \mathbf{Ex}^a \neq \emptyset$.

Theorem 15 implies the following hierarchy results ((3) was already shown in [21]).

Corollary 16. *For every* $a \in \mathbb{N}$,

(1) $\mathbf{RefEx}^a \subset \mathbf{RefEx}^{a+1}$,
(2) $\mathbf{WRefEx}^a \subset \mathbf{WRefEx}^{a+1}$,
(3) $\mathbf{RelEx}^a \subset \mathbf{RelEx}^{a+1}$.

Now a proof similar to the proof of Theorem 15 can be used to show the following result. Notice that $\mathbf{Ex}^* \setminus \bigcup_{a \in \mathbb{N}} \mathbf{Ex}^a \neq \emptyset$ was proved in [10].

Theorem 17. $\mathbf{RefEx}^* \setminus \bigcup_{a \in \mathbb{N}} \mathbf{Ex}^a \neq \emptyset$.

Theorem 15 implies further corollaries. In [10], $\mathbf{Ex}^* \subseteq \mathbf{Bc}$ was shown. This result extends to all our types of refutable learning.

Proposition 8. *For* $\mathbf{I} \in \{\mathbf{Ref}, \mathbf{WRef}, \mathbf{Rel}\}$, $\mathbf{IEx}^* \subseteq \mathbf{IBc}$.

In [10] it was proved that $\mathbf{Bc} \setminus \mathbf{Ex}^* \neq \emptyset$. This result holds refutably.

Corollary 18. $\mathbf{RefBc} \setminus \mathbf{Ex}^* \neq \emptyset$.

The next corollary points out that already $\mathbf{RefEx}^1$ contains "algorithmically rich" classes of *predicates*.

Corollary 19. $\mathbf{RefEx}^1 \cap 2^{\mathcal{R}_{0,1}} \not\subseteq \mathbf{NUM} \cap 2^{\mathcal{R}_{0,1}}$.

Corollary 19 can be even strengthened by replacing $\mathbf{RefEx}^1$ with $\mathbf{RefEx}$. This another time exhibits the richness of already the most stringent of our types of learning refutably.

Theorem 20. $\mathbf{RefEx} \cap 2^{\mathcal{R}_{0,1}} \not\subseteq \mathbf{NUM} \cap 2^{\mathcal{R}_{0,1}}$.

Note that Theorem 20 contrasts a known result on reliable **Ex**-learning. If we require the **Ex**-learning machine's reliability not only on $\mathcal{R}$, but even on the set of all *total* functions, then all classes of recursive *predicates* belonging to this latter type are in **NUM**, see [14].

We now give the analogue to Theorem 15 for $\mathbf{Bc}^a$-learning rather than $\mathbf{Ex}^a$-learning. Note that $\mathbf{Bc}^{a+1} \setminus \mathbf{Bc}^a \neq \emptyset$ was shown in [10].

Theorem 21. *For any* $a \in \mathbb{N}$, $\mathbf{RefBc}^{a+1} \setminus \mathbf{Bc}^a \neq \emptyset$.

Theorem 21 yields the following hierarchies, where (3) solves an open problem from [22].

Corollary 22. *For every* $a \in \mathbb{N}$,

(1) $\mathbf{RefBc}^a \subset \mathbf{RefBc}^{a+1}$,
(2) $\mathbf{WRefBc}^a \subset \mathbf{WRefBc}^{a+1}$,
(3) $\mathbf{RelBc}^a \subset \mathbf{RelBc}^{a+1}$.

Theorem 23. $\mathbf{RefBc}^* \setminus \bigcup_{a \in \mathbb{N}} \mathbf{Bc}^a \neq \emptyset$.

In the proof of Theorem 3 we have derived that *FINSUP* $\notin \mathbf{WRefEx}$. This result is now strengthened for $\mathbf{WRefBc}^a$-learning and then used in the next corollary below.

Theorem 24. *For every* $a \in \mathbb{N}$, *FINSUP* $\notin \mathbf{WRefBc}^a$.

The next corollary points out the relative strength of **RelEx**-learning over $\mathbf{WRefBc}^a$-learning. In other words, in general, one cannot compensate a stricter refutability constraint by a more liberal learning criterion.

Corollary 25. *For all* $a \in \mathbb{N}$, $\mathbf{RelEx} \setminus \mathbf{WRefBc}^a \neq \emptyset$.

Our final result exhibits the strength of **WRefEx**-learning over $\mathbf{RefBc}^a$-learning. Thus, it is in the same spirit as Corollary 25 above.

Theorem 26. *For all* $a \in \mathbb{N}$, $\mathbf{WRefEx} \setminus \mathbf{RefBc}^a \neq \emptyset$.

Note that Theorems 14, 24 and 26, and Corollary 25 hold even if we replace $\mathbf{Bc}^a$ by any criterion of learning for which Proposition 7 holds.

References

1. D. Angluin and C. Smith. Inductive inference: Theory and methods. *Computing Surveys*, 15:237–289, 1983.

2. J. Bārzdiņš. Two theorems on the limiting synthesis of functions. In *Theory of Algorithms and Programs, Vol. 1*, pp. 82–88. Latvian State University, 1974. In Russian.
3. J. Bārzdiņš and R. Freivalds. Prediction and limiting synthesis of recursively enumerable classes of functions. *Latvijas Valsts Univ. Zimatm. Raksti*, 210:101–111, 1974.
4. S. Ben-David. Can finite samples detect singularities of real-valued functions? In *24th Annual ACM Symposium on the Theory of Computing*, pp. 390–399, 1992.
5. L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Inform. and Control*, 28:125–155, 1975.
6. M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.
7. J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.
8. J. Case, S. Jain, and S. Ngo Manguelle. Refinements of inductive inference by Popperian and reliable machines. *Kybernetika*, 30:23–52, 1994.
9. J. Case, E. Kinber, A. Sharma, and F. Stephan. On the classification of computable languages. In *Proc. 14th Symposium on Theoretical Aspects of Computer Science*, Vol. 1200 of *Lecture Notes in Computer Science*, pp. 225–236. Springer, 1997.
10. J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.
11. R. Freivalds. Inductive inference of recursive functions: Qualitative theory. In *Baltic Computer Science*, Vol. 502 of *Lecture Notes in Computer Science*, pp. 77–110. Springer, 1991.
12. R. Freivalds, E. Kinber, and C.H. Smith. On the intrinsic complexity of learning. *Information and Computation*, 123(1):64–71, 1995.
13. E.M. Gold. Language identification in the limit. *Inform. and Control*, 10:447–474, 1967.
14. J. Grabowski. Starke Erkennung. In *Strukturerkennung diskreter kybernetischer Systeme, Teil I*, pp. 168–184. Seminarbericht Nr. 82, Department of Mathematics, Humboldt University of Berlin, 1986.
15. G. Grieser. Reflecting inductive inference machines and its improvement by therapy. In *Algorithmic Learning Theory: 7th International Workshop (ALT '96)*, Vol. 1160 of *Lecture Notes in Artificial Intelligence*, pp. 325–336. Springer, 1996.
16. S. Jain. Learning with refutation. *Journal of Computer and System Sciences*, 57(3):356–365, 1998.
17. S. Jain, E. Kinber, R. Wiehagen and T. Zeugmann. Refutable inductive inference of recursive functions. Schriftenreihe der Institute für Informatik/Mathematik, Serie A, SIIM-TR-A-01-06, Medical University at Lübeck, 2001.
18. S. Jain, D. Osherson, J.S. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory.* MIT Press, Cambridge, Mass., second edition, 1999.
19. K. P. Jantke. Reflecting and self-confident inductive inference machines. In *Algorithmic Learning Theory: 6th International Workshop (ALT '95)*, Vol. 997 of *Lecture Notes in Artificial Intelligence*, pp. 282–297. Springer, 1995.
20. W. Jekeli. *Universelle Strategien zur Lösung induktiver Lernprobleme.* MSc Thesis, Dept. of Computer Science, University of Kaiserslautern, 1997.
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. Kinber and T. Zeugmann. One-sided error probabilistic inductive inference and reliable frequency identification. *Information and Computation*, 92(2):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. Lange and P. Watson. Machine discovery in the presence of incomplete or ambiguous data. In *Algorithmic Learning Theory: 4th International Workshop on Analogical and Inductive Inference (AII '94) and 5th International Workshop on Algorithmic Learning Theory (ALT '94)*, Vol. 872 of *Lecture Notes in Artificial Intelligence*, pp. 438–452. Springer, 1994.
25. R. Lindner. *Algorithmische Erkennung*. Dissertation B, University of Jena, 1972.
26. M. Machtey and P. Young. *An Introduction to the General Theory of Algorithms*. North Holland, New York, 1978.
27. E. Minicozzi. Some natural properties of strong identification in inductive inference. *Theoretical Computer Science*, 2:345–360, 1976.
28. T. Miyahara. Refutable inference of functions computed by loop programs. Technical Report RIFIS-TR-CS-112, Kyushu University, Fukuoka, 1995.
29. Y. Mukouchi and S. Arikawa. Inductive inference machines that can refute hypothesis spaces. In *Algorithmic Learning Theory: 4th International Workshop (ALT '93)*, Vol. 744 of *Lecture Notes in Artificial Intelligence*, pp. 123–136. Springer, 1993.
30. Y. Mukouchi and S. Arikawa. Towards a mathematical theory of machine discovery from facts. *Theoretical Computer Science*, 137:53–84, 1995.
31. K. R. Popper. *The Logic of Scientific Discovery*. Harper and Row, 1965.
32. H. Rice. On completely recursively enumerable classes and their key arrays. *The Journal of Symbolic Logic*, 21:304–308, 1956.
33. H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, 1967. Reprinted by MIT Press in 1987.
34. C.H. Smith, R. Wiehagen, and T. Zeugmann. Classifying predicates and languages. *International Journal of Foundations of Computer Science*, 8(1):15–41, 1997.
35. F. Stephan. On one-sided versus two-sided classification. Technical Report Forschungsberichte Mathematische Logik 25/1996, Mathematical Institute, University of Heidelberg, 1996.
36. R. Wiehagen. Characterization problems in the theory of inductive inference. In *Proc. of the 5th International Colloquium on Automata, Languages and Programming*, Vol. 62 of *Lecture Notes in Computer Science*, pp. 494–508. Springer, 1978.
37. R. Wiehagen and C.H. Smith. Generalization versus classification. *Journal of Experimental and Theoretical Artificial Intelligence*, 7:163–174, 1995.
38. T. Zeugmann. A-posteriori characterizations in inductive inference of recursive functions. *J. of Inform. Processing and Cybernetics (EIK)*, 19:559–594, 1983.