TCS-TR-A-08-32

TCS Technical Report

Learning Recursive Functions

by

Thomas Zeugmann and Sandra Zilles

Division of Computer Science

Report Series A

November 18, 2007

Hokkaido University
Graduate School of
Information Science and Technology

Email: thomas@ist.hokudai.ac.jp

Phone: +81-011-706-7684
Fax: +81-011-706-7684

Contents

Learning Recursive Functions

Thomas Zeugmann
Division of Computer Science
Hokkaido University
N-14, W-9
Sapporo 060-0814, Japan
thomas@ist.hokudai.ac.jp

Sandra Zilles
Department of Computing Science
University of Alberta
Edmonton
Alberta, Canada T6G 2E8
zilles@cs.ualberta.ca

November 18, 2007

Abstract

Studying the learnability of classes of recursive functions has attracted considerable interest for at least four decades. Starting with Gold's (1967) model of learning in the limit, many variations, modifications and extensions have been proposed. These models differ in some of the following: the mode of convergence, the requirements intermediate hypotheses have to fulfill, the set of allowed learning strategies, the source of information available to the learner during the learning process, the set of admissible hypothesis spaces, and the learning goals.

A considerable amount of work done in this field has been devoted to the characterization of function classes that can be learned in a given model, the influence of natural, intuitive postulates on the resulting learning power, the incorporation of randomness into the learning process, the complexity of learning, among others.

On the occasion of Rolf Wiehagen's 60th birthday, the last four decades of research in that area are surveyed, with a special focus on Rolf Wiehagen's work, which has made him one of the most influential scientists in the theory of learning recursive functions.

1. Introduction

Emerging from the pioneering work of Gold [47, 48], Solomonoff [92, 93], Barzdin [15], Thiele [95], Blum and Blum [19], and the work done in Riga [11, 12, 13], inductive inference of recursive functions has fascinated many researchers.

By definition, *inductive inference* is the process of generating hypotheses for describing an unknown object from *finitely many data points* about the unknown object. For example, when exploring a physical phenomenon by performing experiments, a physicist obtains a finite sequence of pairs $(x_0, f(x_0))$, $(x_1, f(x_1)), \dots, (x_n, f(x_n))$.

From these examples the physicist tries to *infer* the law f describing the connection between x and $f(x)$. Usually f is a mathematical expression, a formula, i.e., in a very general scenario an *algorithm* computing the function f. Using more and more examples, the hypothesis on hand may be confirmed or falsified. If it is falsified, usually a new hypothesis is generated.

Many philosophers have studied inductive inference during the last 2000 years, too, and several of their findings and principles have served as philosophical basis of the mathematical theory of inductive inference which in turn shed more light on these findings and principles or has suggested alternatives and refinements (cf., e.g., William of Ockham [78], Freivalds [36], Board and Pitt [21], Popper [84], Case and Smith [27] as well as Klette and Wiehagen [63]).

The mathematical basis for the work presented in this survey goes back to Solomonoff [92, 93] who proposed criteria for selecting a hypothesis explaining given data best, Putnam [85] who anticipated several of the earlier results (though on an informal basis) and Gold [47, 48] who has provided a thorough recursion theoretic basis of inductive inference.

Gold [48] considers inductive inference to be an infinite process. The objects to be inferred are recursive functions. In every step $n = 0, 1, 2, \ldots$ of the inference process the inference algorithm has access to successively growing initial segments $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$ of the graph of the target function. Using these initial segments, the inference algorithm computes hypotheses h_n which are interpreted as numbers of programs in a given computable numbering of (all) partial recursive functions. We refer to such a given numbering as a *hypothesis space*. Usually it is required that the hypothesis space contains a program that is correct for the target function. If $h_n \neq h_{n+1}$, then we say that a *mind change* occurred. The sequence of all hypotheses is required to converge to a correct program for the target function. That is, beyond some point, no further mind change occurs, and the hypothesis repeated from that point on is a program that computes the target function without errors.

The model just described is Gold's [48] *identification in the limit* (cf. Definition 8). Based on identification in the limit, a huge variety of inference models has been proposed and studied. Possible modifications comprise the specification of *correctness*, the mode of convergence, requirements on the intermediate hypotheses output, the set of allowed inference algorithms, the set of admissible hypothesis spaces, and the source of information available, among others.

Nowadays there is a largely developed mathematical theory and many results have found their way into monographs [11, 12, 13, 70], books [79, 55], and surveys [5, 6, 30, 63]. On the one hand, the results obtained have considerably enlarged our understanding of inference processes and learning and their connections to philosophy, cognitive science, psychology, and artificial intelligence. On the other hand, younger counterparts of learning theory and machine learning share with inductive inference several methods, approaches, ideas, techniques and even algorithms and throughout this survey we shall occasionally point to them.

Whenever one tries to survey such a large field, one has to make a certain selection. In the present survey we focused to a larger part on the earlier work done in the field and on research performed by Rolf Wiehagen and his co-workers. An obvious reason for this choice is of course Rolf Wiehagen's 60th birthday which inspired this project. Another aspect has been the availability of the relevant literature and presence and non-existence, respectively, of surveys covering already part of the research undertaken in inductive inference of recursive functions. For example, there are beautiful surveys concerning the learnability of recursive functions via queries (cf. Gasarch and Smith [45]), by teams of inductive inference machines (cf. Smith [89]), or probabilistic inductive inference (cf., e.g., Pitt [81], Ambainis [3]). So, these parts of the theory are only touched in the present paper as is the material presented in Angluin and Smith [5, 6]. Likewise, we had no intention to rewrite the comprehensive paper by Case and Smith [27] which covers many earlier theoretical results of the inductive inference of recursive functions. But of course, some overlapping occasionally occurs.

After introducing some basic notions and notations in Section 2, we start with a list of desiderata seemingly arising naturally when one wishes to define a learning model. In Section 3, we study the resulting learning model, provide different characterizations of it and point to its strengths and weaknesses. We continue with possible alternatives to enlarge the learning power of the first model. This directly leads us to the notion of *consistent learning*. Consistency, which here means that inference algorithms always return hypotheses agreeing with the information they have seen so far, is often presupposed in applications. The question to which extent this affects learning – and the resulting (in)consistency phenomenon – are studied in this survey in more detail (cf. Section 4 and 7).

This study performed in Section 4 as well as the results obtained earlier suggest to introduce further learning models, among them Gold's [48] original model of *learning in the limit* (cf. Section 5). Here we also look at different variations of learning in the limit by changing the mode of convergence, by varying the set of admissible strategies and the information supply.

For gaining a better understanding of the similarities and differences of the various learning types presented so far, we then continue with characterizations in terms of complexity and of computable numberings (cf. Sections 6 and 8, respectively).

While having provided a rather comprehensive treatment of the material mentioned so far, in Section 9 we briefly survey additional research such as learning from good examples, intrinsic complexity and uniformity. The reason we only sketch these areas is the same mentioned above, i.e., there are already comprehensive articles in print that cover these areas. Finally, we provide a summary and discuss open problems.

2. Preliminaries

Unspecified notations follow Rogers [86]. In addition to or in contrast with [86] we use the following. By $\mathbb{N} = \{0, 1, 2, \dots\}$ we denote the set of all natural numbers. We set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. The set of all finite sequences of natural numbers is denoted by $\mathbb{N}^*$.

The cardinality of a set S is denoted by $|S|$. We write $\wp(S)$ for the power set of set S. Let $\emptyset$, $\in$, $\subset$, $\subseteq$, $\supset$, $\supseteq$, and $\#$ denote the empty set, element of, proper subset, subset, proper superset, superset, and incomparability of sets, respectively.

By $\mathfrak{P}$ and $\mathfrak{T}$ we denote the set of all partial and total functions of one variable over $\mathbb{N}$. The set of all partial recursive and recursive functions of one respectively two variables over $\mathbb{N}$ is denoted by $\mathcal{P}$, $\mathcal{R}$, $\mathcal{P}^2$, $\mathcal{R}^2$, respectively. Let $f \in \mathcal{P}$, then we use $\mathtt{dom}(f)$ to denote the *domain* of the function f, i.e., $\mathtt{dom}(f) = \{x \mid x \in \mathbb{N},\ f(x) \text{ is defined}\}$. Additionally, by $\mathtt{Val}(f)$ we denote the *range* of f, i.e., $\mathtt{Val}(f) = \{f(x) \mid x \in \mathtt{dom}(f)\}$. We use $\mathcal{R}_{\{0,1\}}$ to denote the set of all $f \in \mathcal{R}$ satisfying $\mathtt{Val}(f) \subseteq \{0, 1\}$. We refer to $\mathcal{R}_{\{0,1\}}$ as to the set of recursive predicates. A function $f \in \mathcal{P}$ is said to be monotone provided for all $x, y \in \mathbb{N}$ with $x \leqslant y$ we have, if both $f(x)$ and $f(y)$ are defined then $f(x) \leqslant f(y)$. By $\mathcal{R}_{mon}$ we denote the set of all monotone recursive functions.

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. $\psi_i(x)\uparrow$).

For any functions $f, g \in \mathcal{P}$ and any number $m \in \mathbb{N}$ we write $f =_m g$ iff $\{(x, f(x)) \mid x \leqslant m \text{ and } f(x)\downarrow\} = \{(x, g(x)) \mid x \leqslant m \text{ and } g(x)\downarrow\}$; otherwise we write $f \neq_m g$.

A numbering $\varphi \in \mathcal{P}^2$ is called a *Gödel numbering* (cf. Rogers [86]) iff $\mathcal{P}_\varphi = \mathcal{P}$, and for any numbering $\psi \in \mathcal{P}^2$, there is a *compiler* $c \in \mathcal{R}$ such that $\psi_i = \varphi_{c(i)}$ for all $i \in \mathbb{N}$. $\mathtt{Göd}$ denotes the set of all Gödel numberings. Let $\varphi \in \mathtt{Göd}$ and let $f \in \mathcal{P}$; then we use $\min_\varphi f$ to denote the least number i such that $\varphi_i = f$.

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

Following [67] we call any pair (φ, Φ) a *measure of computational complexity* provided φ is a Gödel numbering of $\mathcal{P}$ and $\Phi \in \mathcal{P}^2$ satisfies Blum's [20] axioms. That is, (1) $\mathtt{dom}(\varphi_i) = \mathtt{dom}(\Phi_i)$ for all $i \in \mathbb{N}$ and (2) the predicate "$\Phi_i(x) = y$" is uniformly recursive for all $i, x, y \in \mathbb{N}$. We refer to a measure of computational complexity as to a *complexity measure* for short.

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 $x \leqslant k$, $f(k+1) = j$, and $f(x) = p(x-k-2)$, if $x \geqslant k + 2$. Let $g \in \mathcal{P}$ and $\alpha = (a_0, \ldots, a_k) \in \mathbb{N}^*$; we write $\alpha \sqsubseteq g$ iff α is a prefix of the sequence of values associated with g, i.e., for any $x \leqslant k$, $g(x)$ 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 functions in $\mathcal{U}$. Also, it is convenient to have a notation for the set of all finite variants of functions in $\mathcal{U}$. We use $[[\mathcal{U}]]$ for this set, i.e., $[[\mathcal{U}]] = \{f \mid f \in \mathcal{R},\ \exists f' \in \mathcal{U} \wedge \forall^\infty x[f(x) = f'(x)]\}$. The quantifier $\forall^\infty$, as used here, means "for all but finitely many."

Furthermore, using a fixed encoding $\langle \dots \rangle$ of $\mathbb{N}^*$ onto $\mathbb{N}$ we write f^n instead of $\langle (f(0), \dots, f(n)) \rangle$, for any $n \in \mathbb{N}$, $f \in \mathcal{R}$. Furthermore, the set of all permutations of $\mathbb{N}$ is denoted by $\Pi(\mathbb{N})$. Any element $X \in \Pi(\mathbb{N})$ can be represented by a unique sequence $(x_n)_{n \in \mathbb{N}}$ that contains each natural number precisely once. Let $X \in \Pi(\mathbb{N}), f \in \mathcal{P}$ and $n \in \mathbb{N}$. Then we write $f^{X,n}$ instead of $\langle (x_0, f(x_0), \dots, x_n, f(x_n)) \rangle$ provided $f(x_k)$ is defined for all $k \leqslant n$.

Finally, a sequence $(j_n)_{n \in \mathbb{N}}$ of natural numbers is said to *converge* to the number j iff all but finitely many numbers of it are equal to j. A sequence $(j_n)_{n \in \mathbb{N}}$ of natural numbers is said to *finitely converge* to the number j iff it converges in the limit to j and for all $n \in \mathbb{N}$, $j_n = j_{n+1}$ implies $j_k = j$ for all $k \geqslant n$.

In the following section, we introduce the subject of this survey, i.e., learning of recursive functions. For making this survey self-contained, first we briefly outline what we have to specify in order to arrive at a learning model for recursive functions. Then we provide an important example.

3. Defining a Learning Model

In the following, the learner will be an algorithm. We refer to it as a *strategy* S. That is, we shall require $S \in \mathcal{P}$. The objects to be learned are recursive functions. Thus, the next question we have to address is from what information recursive functions should be learned. The information fed to the strategy are finite lists of pairs "argument-value," i.e., lists $(x_0, f(x_0)), \dots, (x_n, f(x_n))$. So, for technical convenience we describe this information by using the notation $f^{X,n}$ defined above. If the order in which examples are presented does not matter, then we restrict ourselves to present examples in natural order, i.e., we consider lists $(0, f(0))$, $(1, f(1)), \dots, (n, f(n))$. If examples are presented in natural order, the argument is redundant. Thus, we can use the notation f^n defined above to describe the information fed to the strategy.

Additionally, we require that the entirety of the local information completely describes the function f to be learned. That means, for every $n \in \mathbb{N}$ there must be a finite list containing $(n, f(n))$.

Using the local information $f^{X,n}$, the strategy computes a number i which is referred to as a *hypothesis*. Thus, when successively fed the sequence $(f^{X,n})_{n \in \mathbb{N}}$, the strategy computes a sequence of hypotheses which is interpreted with respect to a suitably chosen hypothesis space. Hypothesis spaces are numberings $\psi \in \mathcal{P}^2$ which are required to contain at least one program for every function to be learned.

Finally, we require the sequence of hypotheses formed in the way described above to converge to a program that correctly computes the target function.

Usually, we consider sets $\mathcal{U}$ of recursive functions. Given a class $\mathcal{U} \subseteq \mathcal{R}$ we then have to ask whether or not the resulting learning problem is solvable. For obtaining an affirmative answer we have to provide a strategy S learning every function in $\mathcal{U}$. Otherwise, we have to show that there is *no* strategy S which can learn every function in $\mathcal{U}$.

In order to have some examples, it is useful to define some function classes which we shall use quite often throughout this survey. First, let

$$\mathcal{U}_0 = \{f \mid f \in \mathcal{R} \text{ and } \forall^\infty n[f(n) = 0]\}$$

be the class of all functions that are almost everywhere zero. This class is also known as the class of functions of finite support. It is easy to see that $\mathcal{U}_0 \in \mathcal{NUM}$.

Next, let (φ, Φ) be any fixed complexity measure. We set

$$\mathcal{U}_{(\varphi,\Phi)} = \{\Phi_i \mid \varphi_i \in \mathcal{R}\}$$

and refer to $\mathcal{U}_{(\varphi,\Phi)}$ as to the class of all recursive complexity functions.

Another quite popular class is the class of self-describing functions defined as follows. Let $\varphi \in \mathcal{P}^2$ be any fixed Gödel numbering; we set

$$\mathcal{U}_{sd} = \{f \mid f \in \mathcal{R} \text{ and } \varphi_{f(0)} = f\}\ .$$

Note that neither $\mathcal{U}_{(\varphi,\Phi)}$ nor $\mathcal{U}_{sd}$ belong to $\mathcal{NUM}$ as we shall prove next.

Lemma 1. $\mathcal{U}_{(\varphi,\Phi)},\ \mathcal{U}_{sd} \notin \mathcal{NUM}$

Proof. For showing that $\mathcal{U}_{(\varphi,\Phi)} \notin \mathcal{NUM}$ we first observe that for every class $\mathcal{U} \in \mathcal{NUM}$ there is a function $b \in \mathcal{R}$ such that $\forall^\infty x[f(x) \leqslant b(x)]$ for every function $f \in \mathcal{U}$. This can be seen as follows. Let $\psi \in \mathcal{R}^2$ be such that $\mathcal{U} \subseteq \mathcal{R}_\psi$. Then it suffices to set $b(x) = \max\{\psi_i(x) \mid i \leqslant x\}$. Supposing $\mathcal{U}_{(\varphi,\Phi)} \in \mathcal{NUM}$ there would be such a function b for the class $\mathcal{U}_{(\varphi,\Phi)}$. The desired contradiction is obtained by the following claim.

Claim 1. *Let* $f \in \mathcal{R}$ *be arbitrarily fixed. Then there is a* φ*-program* i *such that* $\varphi_i = f$ *and* $\Phi_i(x) > b(x)$ *for all* $x \in \mathbb{N}$.

Let $s \in \mathcal{R}$ be chosen such that

$$\varphi_{s(j)}(x) = \begin{cases} f(x), & \text{if } \neg[\Phi_j(x) \leqslant b(x)] \\ \varphi_j(x) + 1, & \text{if } \Phi_j(x) \leqslant b(x)\ . \end{cases}$$

By the fixed point theorem (cf., e.g., Smith [88]) there is a number i such that $\varphi_{s(i)} = \varphi_i$. Suppose there is an x such that $\Phi_i(x) \leqslant b(x)$. By construction $\varphi_i(x) = \varphi_{s(i)}(x) = \varphi_i(x) + 1$, a contradiction. So, this case cannot happen and we get $\varphi_i = \varphi_{s(i)} = f$. This proves the claim.

Consequently, $\mathcal{U}_{(\varphi,\Phi)} \notin \mathcal{NUM}$.

In order to show that $\mathcal{U}_{sd} \notin \mathcal{NUM}$ we first prove that $\mathcal{R} \notin \mathcal{NUM}$. Suppose the converse, i.e., there is a numbering $\psi \in \mathcal{R}^2$ such that $\mathcal{R} \subseteq \mathcal{R}_\psi$. We define a function f by setting $f(x) = \psi_x(x) + 1$ for all $x \in \mathbb{N}$. Since $\psi \in \mathcal{R}^2$ we obtain $f \in \mathcal{R}$. Hence, there should be a ψ-program for f, say j, i.e., $\psi_j = f$. But $\psi_j(j) = f(j) = \psi_j(j) + 1$, a contradiction. So we have $\mathcal{R} \notin \mathcal{NUM}$.

Now the proof of $\mathcal{U}_{sd} \notin \mathcal{NUM}$ is obtained by the following claim.

Claim 2. *For every* $f \in \mathcal{R}$ *there is an* $i \in \mathbb{N}$ *such that* $\varphi_i(0) = i$ *and* $\varphi_i(x+1) = f(x)$ *for all* $x \in \mathbb{N}$.

Let $f \in \mathcal{R}$ be any function and let $s \in \mathcal{R}$ be chosen such that for all $j \in \mathbb{N}$

$$\varphi_{s(j)}(x) = \begin{cases} j, & \text{if } x = 0 \\ f(x-1), & \text{if } x > 0 \ . \end{cases}$$

Again, by the fixed point theorem there is a number i such that $\varphi_{s(i)} = \varphi_i$. By construction, $\varphi_i(0) = i$ and $\varphi_i(x+1) = f(x)$ for all $x \in \mathbb{N}$. This proves Claim 2.

Now, if $\mathcal{U}_{sd} \in \mathcal{NUM}$, then, by erasing the first argument, one can directly obtain a numbering ψ such that $\mathcal{R} = \mathcal{R}_\psi$, a contradiction to $\mathcal{R} \notin \mathcal{NUM}$. ∎

The following classes are due to Blum and Blum [19]. Let (φ, Φ) be any complexity measure, and let $\tau \in \mathcal{R}$ be such that for all $i \in \mathbb{N}$

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

Now we set

$$\mathcal{U}_{mahp} = \{\varphi_{\tau(i)} \mid i \in \mathbb{N} \text{ and } \Phi_i \in \mathcal{R}_{mon}\}$$

(the class of *monotone approximations to the halting problem*) and

$$\mathcal{U}_{ahp} = \{\varphi_{\tau(i)} \mid i \in \mathbb{N} \text{ and } \Phi_i \in \mathcal{R}\}$$

(the class of *approximations to the halting problem*).

Note that $\mathcal{U}_{mahp}$, $\mathcal{U}_{ahp} \notin \mathcal{NUM}$. For a proof, we refer the reader to Stephan and Zeugmann [94].

Whenever appropriate, we shall use these function classes for illustration of the learning models defined below, by analyzing whether or not the corresponding learning problem is solvable.

Next, we exemplify the definition of a learning model and characterize the collection of all function classes $\mathcal{U}$ for which the learning problem is solvable.

3.1. *The Learning Types* $\mathcal{R}$-$\mathcal{TOTAL}^{arb}$ *and* $\mathcal{R}$-$\mathcal{TOTAL}$

Let us start with a list of desiderata. First, we do not make any assumption concerning the order in which examples are presented. Second, our strategy should be defined on all inputs, i.e., we require $S \in \mathcal{R}$. This may seem convenient, since it may be hard to know which inputs to the strategy may occur. Third, every hypothesis should describe a recursive function. Again, this looks natural, since any hypothesis not describing a recursive function cannot be correct. Thus allowing a strategy to output hypotheses not describing recursive functions may be a source of potential errors which we avoid by our requirement. Moreover, this requirement is also nicely in line with Popper's [84] refutability principle requiring that we should be able to refute every incorrect hypothesis.

Definition 1 (Wiehagen [99]). *Let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\psi \in \mathcal{P}^2$. *The class* $\mathcal{U}$ *is said to be* $\mathcal{R}$–totally arb-learnable with respect to ψ *if there is a strategy* $S \in \mathcal{R}$ *such that*

(1) $\psi_{S(n)} \in \mathcal{R}$ *for all* $n \in \mathbb{N}$,

(2) *for all* $f \in \mathcal{U}$ *and every* $X \in \Pi(\mathbb{N})$, *there is a* $j \in \mathbb{N}$ *such that* $\psi_j = f$, *and* $(S(f^{X,n}))_{n\in\mathbb{N}}$ *converges to* j.

If $\mathcal{U}$ *is* $\mathcal{R}$*–totally arb-learnable with respect to* ψ *by a strategy* S, *we write* $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}^{arb}_{\psi}(S)$. *Moreover, let* $\mathcal{R}\text{-}\mathcal{TOTAL}^{arb}_{\psi} = \{\mathcal{U} \mid \mathcal{U}$ is $\mathcal{R}$–totally arb-learnable w.r.t. $\psi\}$, *and let* $\mathcal{R}\text{-}\mathcal{TOTAL}^{arb} = \bigcup_{\psi\in\mathcal{P}^2} \mathcal{R}\text{-}\mathcal{TOTAL}^{arb}_{\psi}$.

Some remarks are mandatory here. Let us start with the semantics of the hypotheses produced by a strategy S. As described above, we always interpret the number $S(f^{X,n})$ as a ψ–number. This convention is adopted to all the definitions below. The "arb" in arb-learnable points to the fact that we require learnability with respect to any arbitrary order of the input. Moreover, according to the definition of convergence, only finitely many data points of the graph of a function f were available to the strategy S up to the unknown point of convergence. Therefore, some form of learning must have taken place. Thus, the use of the term "learn" in the above definition is indeed justified.

Note that $\mathcal{R}\text{-}\mathcal{TOTAL}^{arb}$ is sometimes also called PEX, where the EX stands for *explain* and P refers to *Popperian* strategies, i.e., strategies that can directly use Popper's [84] refutability principle (cf. [27]). But we think this interpretation of Popper's [84] refutability principle is too narrow. A more detailed discussion is provided throughout this survey.

In order to study the impact of the requirement to learn with respect to any order of the input, next we relax Definition 1 by demanding only learnability from input presented in natural order.

Definition 2 (Wiehagen [99]). *Let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\psi \in \mathcal{P}^2$. *The class* $\mathcal{U}$ *is said to be* $\mathcal{R}$–totally learnable with respect to ψ *if there is a strategy* $S \in \mathcal{R}$ *such that*

(1) $\psi_{S(n)} \in \mathcal{R}$ *for all* $n \in \mathbb{N}$,

(2) *for each* $f \in \mathcal{U}$ *there is a* $j \in \mathbb{N}$ *such that* $\psi_j = f$, *and* $(S(f^n))_{n\in\mathbb{N}}$ *converges to* j.

$\mathcal{R}\text{-}\mathcal{TOTAL}_{\psi}(S)$, $\mathcal{R}\text{-}\mathcal{TOTAL}_{\psi}$, *and* $\mathcal{R}\text{-}\mathcal{TOTAL}$ *are defined analogously to the above.*

It is technically advantageous to start with the following result showing that, as far as $\mathcal{R}$–total learning is concerned, the order in which the graph of the function is fed to the learning strategy does not matter.

Theorem 1. $\mathcal{R}\text{-}\mathcal{TOTAL} = \mathcal{R}\text{-}\mathcal{TOTAL}^{arb}$

Proof. Obviously, if we can learn from arbitrary input then we can learn from input presented in natural order, i.e., $\mathcal{R}\text{-}\mathcal{TOTAL}^{arb} \subseteq \mathcal{R}\text{-}\mathcal{TOTAL}$.

For the opposite direction, let $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}$. Hence there is a numbering $\psi \in \mathcal{P}^2$ and a strategy $S \in \mathcal{R}$ such that $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}_\psi(S)$. The desired strategy S' is obtained from S by adding a preprocessing. If S' receives an encoded list $f^{X,n}$ it looks for the largest number m such that $(0, f(0)), \ldots, (m, f(m))$ are all present in $f^{X,n}$. If this number m exists, then S' simulates S on input f^m and outputs the hypothesis computed. Otherwise, i.e., if $(0, f(0))$ does not occur in $f^{X,n}$, then S' simply returns a fixed program of the constant zero function as an initial auxiliary hypothesis.

Now it is easy to see that $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}_\psi^{arb}(S')$. We omit the details. ∎

The following lemma is both of technical and of epistemological importance. It actually states that, if we can $\mathcal{R}$-totally learn with respect to some numbering, then we can also learn with respect to any Gödel numbering. As we shall see later, its proof directly transforms to almost every learning type considered in this survey.

Lemma 2. *Let* $\mathcal{U} \subseteq \mathcal{R}$*, let* $\psi \in \mathcal{P}^2$ *be any numbering and let* $S \in \mathcal{R}$ *be such that* $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}_\psi(S)$*. Furthermore, let* $\varphi \in \mathcal{P}^2$ *be any Gödel numbering. Then there is a strategy* $\hat{S} \in \mathcal{R}$ *such that* $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}_\varphi(\hat{S})$*.*

Proof. By the definition of a Gödel numbering there is a compiler function $c \in \mathcal{R}$ such that $\psi_i = \varphi_{c(i)}$ for all $i \in \mathbb{N}$. Thus, we can define $\hat{S}(f^n) = c(S(f^n))$ and the lemma follows. ∎

Expressed differently, we have just shown that $\mathcal{R}\text{-}\mathcal{TOTAL} = \mathcal{R}\text{-}\mathcal{TOTAL}_\varphi$ for every Gödel numbering φ. But it is often advantageous to use special numberings having special properties facilitating learning. A first example is provided by Theorem 2 below. Additionally, this theorem also characterizes the classes in $\mathcal{R}\text{-}\mathcal{TOTAL}$.

Theorem 2. $\mathcal{R}\text{-}\mathcal{TOTAL} = \mathcal{NUM}$

Proof. The proof is done by showing two claims.

Claim 1. $\mathcal{R}\text{-}\mathcal{TOTAL} \subseteq \mathcal{NUM}$

Let $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}$. Then there is a strategy $S \in \mathcal{R}$ and a numbering $\psi \in \mathcal{P}^2$ such that $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}_\psi(S)$. We have to construct a numbering $\tau \in \mathcal{R}^2$ such that $\mathcal{U} \subseteq \mathcal{R}_\tau$.

For all $i, x \in \mathbb{N}$ we define $\tau(i, x) = \psi_{S(i)}(x)$. By Condition (1) of Definition 2 we know that $\psi_{S(i)} \in \mathcal{R}$. Thus, we directly obtain $\tau \in \mathcal{R}^2$. It remains to show that $\mathcal{U} \subseteq \mathcal{R}_\tau$. Let $f \in \mathcal{U}$. By Condition (2) of Definition 2 there exists a j such that $\psi_j = f$ and $(S(f^n))_{n\in\mathbb{N}}$ converges to j. Let k be minimal such that $S(f^n) = j$ for all $n \geqslant k$. Thus, for $i = f^k$ we obtain

$$\tau_i = \psi_{S(i)} = \psi_{S(f^k)} = \psi_j = f\ ,$$

and consequently, $\mathcal{U} \subseteq \mathcal{R}_\tau$. This proves Claim 1.

Claim 2. $\mathcal{NUM} \subseteq \mathcal{R}\text{-}\mathcal{TOTAL}$

Let $\mathcal{U} \in \mathcal{NUM}$. Hence there is a numbering $\psi \in \mathcal{R}^2$ such that $\mathcal{U} \subseteq \mathcal{R}_\psi$. Essentially, Claim 2 is proved by using Gold's [48] famous identification by enumeration strategy. The idea behind the identification by enumeration strategy to learn a function $f \in \mathcal{U}$ is to search for the least index j in the enumeration $\psi_0, \psi_1, \psi_2, \ldots$ such that $\psi_j = f$. So on input f^n one looks for the least i such that $\psi_i^n = f^n$.

The only difficulty we have to overcome is to ensure that S satisfies Condition (1) of Definition 2 for all $f \in \mathcal{R}$, that is, also in case $f \in \mathcal{R} \setminus \mathcal{U}$. Then there may be no program i at all such that $\psi_i^n = f^n$.

Therefore, using a fixed enumeration of $\mathbb{N}^*$ (cf. Rogers [86]) we define a numbering χ as follows. Let α be the ith tuple of $\mathbb{N}^*$ enumerated. We set $\chi_i = \alpha 0^\infty$. Thus, $\chi \in \mathcal{R}^2$ and $\mathcal{U}_0 = \mathcal{R}_\chi$.

Next, we define a numbering $\tau \in \mathcal{R}^2$ by setting $\tau_{2i} = \psi_i$ and $\tau_{2i+1} = \chi_i$ for all $i \in \mathbb{N}$. Now, taking into account that $[\mathcal{U}_0] = [\mathcal{R}] = \mathbb{N}^*$, we can directly use the identification by enumeration strategy by using the numbering τ to $\mathcal{R}$-totally learn the class $\mathcal{U}$. This proves Claim 2.

Claim 1 and Claim 2 together yield the theorem. ∎

On the one hand, $\mathcal{NUM}$ is a rich collection of function classes. As a matter of fact, the class of all primitive recursive functions is in $\mathcal{NUM}$. Moreover, the characterization obtained by Theorem 2 directly allows a very strong corollary, which first requires the following simple definition.

Definition 3. *Let* LT *be any learning type and let* $(S_i)_{i \in \mathbb{N}}$ *be a recursive enumeration of strategies fulfilling the requirements of the learning type* LT. *We call* LT closed under recursively enumerable union *if there is a strategy* S *fulfilling the requirements of* LT *such that* $\bigcup_{i \in \mathbb{N}} LT(S_i) \subseteq LT(S)$.

Corollary 3. $\mathcal{R}\text{-}\mathcal{TOTAL}$ *is closed under recursively enumerable union.*

On the other hand, none of the classes $\mathcal{U}_{(\varphi,\Phi)}$, $\mathcal{U}_{sd}$, $\mathcal{U}_{mahp}$, and $\mathcal{U}_{ahp}$ is in $\mathcal{NUM}$ as pointed out above.

So, we have to explore some ways to enlarge the learning capabilities of $\mathcal{R}\text{-}\mathcal{TOTAL}$. Before doing this, we also characterize $\mathcal{R}\text{-}\mathcal{TOTAL}$ in terms of complexity, since it may help to gain a better understanding of the properties making a function class learnable or non-learnable, respectively.

The idea behind the following characterization can be explained easily. Suppose we want to learn a class $\mathcal{U}$ with respect to any fixed Gödel numbering φ. Then a strategy may try to find a program i such that $\varphi_i^n = f^n$. Though this search will succeed, the strategy may face serious difficulties to converge. To see this, suppose on input f^n a program i as described has been found. Next, the strategy sees also $f(n+1)$. Now it may try to compute $\varphi_i(n+1)$ and, in parallel to find again an index, say j, such that $\varphi_j^{n+1} = f^{n+1}$. Once j is found and the computation of $\varphi_i(n+1)$

has not stopped yet, the strategy must make a decision. Either it tries to compute $\varphi_i(n+1)$ further or it switches its hypothesis to j. The latter would be a bad idea if $\varphi_j \neq f$ but $\varphi_i = f$. On the other hand, it would be a good idea if $\varphi_i(n+1)\uparrow$. Since the halting problem is undecidable, without any additional information, the strategy cannot decide which case actually occurs. Thus, it is intuitively clear that information concerning the computational complexity of the functions to be learned can only help. We illustrate this by reproving Barzdin's and Freivalds' [14] Extrapolation Theorem here in our setting.

Let $t \in \mathcal{R}$, and let (φ, Φ) be any fixed complexity measure. Following McCreight and Meyer [74], we define the complexity class

$$\mathcal{R}_t = \{\varphi_i \mid \forall^\infty n[\Phi_i(n) \leqslant t(n)]\} \cap \mathcal{R} .$$

For further information concerning these complexity classes, we refer the interested reader to e.g., [22, 28, 67, 105].

Theorem 4 (Barzdin and Freivalds [14]). *For every class* $\mathcal{U} \subseteq \mathcal{R}$ *we have:* $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}$ *if and only if there is a function* $t \in \mathcal{R}$ *such that* $\mathcal{U} \subseteq \mathcal{R}_t$.

Proof. Necessity. Let $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{TOTAL}$. Then, by Theorem 2, we know that there is a numbering $\psi \in \mathcal{R}^2$ such that $\mathcal{U} \subseteq \mathcal{R}_\psi$. Now let $c \in \mathcal{R}$ be any fixed compiler such that $\psi_i = \varphi_{c(i)}$ for all $i \in \mathbb{N}$. We set $t(n) = \max\{\Phi_{c(i)}(n) \mid i \leqslant n\}$. Clearly, $\mathcal{U} \subseteq \mathcal{R}_t$.

Sufficiency. Suppose $\mathcal{U} \subseteq \mathcal{R}_t$. By Theorem 2, it suffices to show that $\mathcal{R}_t \in \mathcal{NUM}$. For proving this, we use the observation that $f \in \mathcal{R}_t$ if and only if there are j, n, $k \in \mathbb{N}$ such that $f = \varphi_j$, $\Phi_j(x) \leqslant k$ for all $x \leqslant n$ and $\Phi_j(x) \leqslant t(x)$ for all $x > n$. Now let c_3 be the canonical enumeration of $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$. For $c_3(i) = (j, n, k)$ and $x \in \mathbb{N}$ we define

$$\psi(i, x) = \begin{cases} \varphi_j(x), & \text{if } x \leqslant n \text{ and } \Phi_j(x) \leqslant k \\ \varphi_j(x), & \text{if } x > n \text{ and } \Phi_j(x) \leqslant t(x) \\ 0, & \text{otherwise.} \end{cases}$$

By construction, we clearly have $\psi \in \mathcal{R}^2$. Now let $f \in \mathcal{R}_t$. Using the observation made above, choose i such that $c_3(i) = (j, n, k)$, where $f = \varphi_j$, $\Phi_j(x) \leqslant k$ for all $x \leqslant n$ and $\Phi_j(x) \leqslant t(x)$ for all $x > n$. Hence, $\psi_i = \varphi_j = f$ and thus $f \in \mathcal{R}_\psi$. Consequently, $\mathcal{R}_t \in \mathcal{NUM}$. ∎

There is another nice characterization of $\mathcal{R}\text{-}\mathcal{TOTAL}$ in terms of a different learning model which we would like to include. First we define the learning model which has been introduced by Barzdin [15].

Definition 4 (Barzdin [15]). *A class* $\mathcal{U} \subseteq \mathcal{R}$ *of functions is said to be* predictable *if there exists a strategy* $S \in \mathcal{R}$ *such that* $S(f^n) = f(n+1)$ *for all* $f \in \mathcal{U}$ *and all but finitely many* $n \in \mathbb{N}$.

The resulting learning type is denoted by $\mathcal{NV}$. Here, $\mathcal{NV}$ stands for "next-value." So, in $\mathcal{NV}$ learning we have to correctly predict the next value of the target function for almost all n.

Theorem 5 (Barzdin [15]). $\mathcal{NV} = \mathcal{R}\text{-}\mathcal{TOTAL}$

We do not prove this theorem here but refer the interested reader to Case and Smith [27] (cf. Theorem 2.19).

But we would like to discuss another interesting aspect. If the value predicted by an $\mathcal{NV}$ learner is wrong, i.e., if $S(f^n) \neq f(n+1)$, then we say that a *prediction error* occurs. Analogously, if an $\mathcal{R}\text{-}\mathcal{TOTAL}$ learner changes its hypothesis, i.e., if $S(f^n) \neq S(f^{n+1})$, then S performs a *mind change*.

Now, when using the identification by enumeration strategy, in order to learn the nth function enumerated in the numbering ψ, one needs n mind changes in the worst case and this approach also leads to n prediction errors in the worst case. Therefore, it is only natural to ask whether or not we can do any better. In fact, an exponential speed-up is possible. For the sake of simplicity, we present the solution here only for classes of recursive predicates, i.e., $\mathcal{U} \subseteq \mathcal{R}_{\{0,1\}}$ and for prediction errors.

Theorem 6 (Barzdin and Freivalds [17]). *Let $\psi \in \mathcal{R}^2$ such that $\psi_i \in \mathcal{R}_{\{0,1\}}$ for all $i \in \mathbb{N}$. Then there exists an $\mathcal{NV}$ learner $\mathcal{FP}$ for $\mathcal{U}$ making at most $O(\log n)$ prediction errors for every function $f \in \mathcal{U}$, where n is the least number j such that $\psi_j = f$.*

Proof. Let $f \in \mathcal{U}$ be the target function. The desired $\mathcal{NV}$ learner works in stages. In each Stage i it considers the subset of the block of functions $B_i = \{\psi_k \mid 2^{2^i} + 1 \leqslant k \leqslant 2^{2^{i+1}}\}$ that coincide with all the data seen so far. Then it makes its prediction in accordance with the majority of the functions still in the block. After having read the true value, it eliminates the functions not coinciding with the new value from block B_i. If all functions are eventually eliminated, Stage i is left and Stage $i+1$ is started. Clearly, if the target function f belongs to block B_i, Stage i is never left. Before analyzing this prediction algorithm we give a formal description of it. In order to make it better readable, we also add the arguments to the data presentation.

Algorithm $\mathcal{FP}$: "On successive input $\langle 0, f(0), 1, f(1), 2, f(2), \ldots \rangle$ do the following:
Execute Stage 0:

Stage 0: Set $V_0 = \{0, 1, 2, 3, 4\}$, $x_0 = 0$.

While $V_0 \neq \emptyset$ execute (A) else goto Stage 1.

(A) Read x_0. Compute $V_0^0 = \{k \mid k \in V_0,\ \psi_k(x_0) = 0\}$, and
$V_0^1 = \{k \mid k \in V_0,\ \psi_k(x_0) = 1\}$.
If $|V_0^0| \geqslant |V_0^1|$ then predict 0; otherwise predict 1.
Read $f(x_0)$, and increment x_0. If $f(x_0) = 0$ set $V_0 = V_0^0$; otherwise set $V_0 = V_0^1$.

Stage i, $i \geqslant 1$: Set $x_i = x_{i-1}$, and compute
$V_i = \{k \in \mathbb{N} \mid 2^{2^i} + 1 \leqslant k \leqslant 2^{2^{i+1}},\ \psi_k(x) = f(x) \text{ for all } 0 \leqslant x \leqslant x_i\}$.

(* V_i is the set of those indices of functions in block i that coincide with all the data seen so far. *)

While $V_i \neq \emptyset$ execute (B) else goto Stage $i + 1$.

(B) Read x_i. Compute $V_i^0 = \{k \mid k \in V_i,\ \psi_k(x_i) = 0\}$, and $V_i^1 = \{k \mid k \in V_i,\ \psi_k(x_i) = 1\}$.
If $|V_i^0| \geqslant |V_i^1|$ then predict 0; otherwise predict 1.
Read $f(x_i)$, and increment x_i. If $f(x_i) = 0$ set $V_i = V_i^0$; otherwise set $V_i = V_i^1$.

We start our analysis by asking how many stages the algorithm $\mathcal{FP}$ has to execute. Let n be the least number j such that $\psi_j = f$. Furthermore, let i be the least number m such that $n \in V_m$. Thus, $i = \lceil \log\log n \rceil - 1$. The total number of prediction mistakes is the sum of all the prediction mistakes made on each of the blocks $V_0, V_1, \ldots V_i$. The number of prediction mistakes made on V_0 is 3. For every $1 \leqslant z < i$ the number of prediction mistakes made on V_z will be at most $\lceil \log(|V_z|) \rceil$. To see this, remember that each prediction is made in accordance with the majority of computed values for all the remaining indices in V_z. Thus, whenever a prediction error occurs, at least half of the indices in V_z is deleted. Since all indices are eventually deleted, we arrive at the stated bound. Analogously, the number of prediction mistakes made on V_i is at most $\lceil \log |V_i| \rceil - 1$. Obviously, $|V_z| = 2^{2^z}(2^{2^z} - 1)$, and thus $\lceil \log |V_z| \rceil \leqslant 2^{z+1}$.

Therefore, the maximum number of prediction mistakes is upper bounded by

$$\begin{aligned} 2^1 + \cdots + 2^{i+1} &= 2^{i+2} - 1 \\ &\leqslant 2^{\lceil \log\log n \rceil - 1 + 2} - 1 \\ &\leqslant 4 \cdot 2^{\log\log n} \\ &= 4 \log n \\ &= O(\log n)\ . \quad \blacksquare \end{aligned}$$

The algorithm $\mathcal{FP}$ invented by Barzdin and Freivalds is nowadays usually referred to as the *halving algorithm* which has found many applications in machine learning. This algorithm can be modified to totally learn every class of recursive predicates with at most $O(\log n)$ mind changes. However, in order to achieve this result, the resulting strategy must use a Gödel numbering as its hypothesis space and not the numbering ψ. Furthermore, all these results can be generalized to learn or to predict arbitrary classes from $\mathcal{NUM}$, thereby still achieving the $O(\log n)$ bound. For a detailed presentation and further information, we refer the reader to Freivalds, Bārzdiņš and Podnieks [37].

The results obtained so far provide some insight concerning the problem how to extend the learning capabilities of $\mathcal{R}$-$\mathcal{TOTAL}$. First, we could restrict our demands to the strategy to hold only on initial segments from $[\mathcal{U}]$ instead of from $[\mathcal{R}]$. Second we could modify our demands to the intermediate hypotheses. The demand to output only programs computing recursive functions seems rather strong.

Third, we could have a closer look at the identification by enumeration strategy. The most obvious point here is that we do not need the requirement $\psi_i \in \mathcal{R}$. For example, if the predicate "$\psi_i(x) = y$" was uniformly recursive for all $i, x, y \in \mathbb{N}$ it would still work. But as we shall see, there is more we can do.

Fourth, looking at the definition of the complexity class $\mathcal{R}_t$, we see that the bound t does not depend on the functions f to be learned. So, some modifications are suggesting themselves.

We finish this chapter, by trying the first approach. The other modifications are discussed later. So, let us relax the definition of $\mathcal{R}\text{-}\mathcal{TOTAL}$ as described above.

Definition 5 (Freivalds and Barzdin [34]). *Let $\mathcal{U} \subseteq \mathcal{R}$ and let $\psi \in \mathcal{P}^2$. The class $\mathcal{U}$ is said to be* totally learnable with respect to ψ *if there is a strategy $S \in \mathcal{P}$ such that for each function $f \in \mathcal{U}$,*

(1) *for all $n \in \mathbb{N}$, $S(f^n)$ is defined and $\psi_{S(f^n)} \in \mathcal{R}$,*

(2) *there is a $j \in \mathbb{N}$ such that $\psi_j = f$, and $(S(f^n))_{n\in\mathbb{N}}$ converges to j.*

$\mathcal{TOTAL}_\psi(S)$, $\mathcal{TOTAL}_\psi$ and $\mathcal{TOTAL}$ are defined analogously as above.

Note that any strategy that learns in the sense of $\mathcal{TOTAL}$ can directly use Popper's [84] refutability principle. But obviously, $\mathcal{U}_{sd} \in \mathcal{TOTAL}$ and thus total learning is more powerful than $\mathcal{R}$-total inference.

Theorem 7. $\mathcal{R}\text{-}\mathcal{TOTAL} \subset \mathcal{TOTAL}$

But the price paid is rather high, since, in contrast to Corollary 3, now we can easily prove that $\mathcal{TOTAL}$ is not closed under union.

Theorem 8. $\mathcal{U}_0 \cup \mathcal{U}_{sd} \notin \mathcal{TOTAL}$

Proof. Suppose the converse. Then there must exist a strategy S such that $\mathcal{U}_0 \cup \mathcal{U}_{sd} \in \mathcal{TOTAL}(S)$. Since $[\mathcal{U}_0] = [\mathcal{R}]$, we can conclude $S \in \mathcal{R}$ and $\varphi_{S(i)} \in \mathcal{R}$ for all $i \in \mathbb{N}$. Hence, S would witness $\mathcal{U}_0 \cup \mathcal{U}_{sd} \in \mathcal{R}\text{-}\mathcal{TOTAL}(S)$. So, by Theorem 2, we obtain $\mathcal{U}_0 \cup \mathcal{U}_{sd} \in \mathcal{NUM}$, a contradiction to Lemma 1. ∎

$\mathcal{TOTAL}$ has another interesting property. Modifying Definition 5 in the opposite way we have obtained $\mathcal{R}\text{-}\mathcal{TOTAL}$ from $\mathcal{R}\text{-}\mathcal{TOTAL}^{arb}$, we get the learning type $\mathcal{TOTAL}^{arb}$. Then, using the same ideas as in the proof of Theorem 1, one can easily show the following theorem first announced in Jantke and Beick [59].

Theorem 9. $\mathcal{TOTAL} = \mathcal{TOTAL}^{arb}$

Next, we characterize $\mathcal{TOTAL}$ in terms of computable numberings.

Theorem 10 (Wiehagen [99]). *Let $\mathcal{U} \subseteq \mathcal{R}$. Then we have: $\mathcal{U} \in \mathcal{TOTAL}$ if and only if there exists a numbering $\psi \in \mathcal{P}^2$ such that*

(1) $\mathcal{U} \subseteq \mathcal{P}_\psi$,

(2) *There is a function* $g \in \mathcal{R}$ *such that* $\psi_i =_{g(i)} f$ *implies* $\psi_i \in \mathcal{R}$ *for every function* $f \in \mathcal{U}$ *and every program* i.

Proof. Necessity. Let $\mathcal{U} \in \mathcal{TOTAL}$ and let $\varphi \in \mathsf{Göd}$. By Lemma 2 we can assume that there is a strategy $S \in \mathcal{P}$ such that $\mathcal{U} \in \mathcal{TOTAL}_\varphi(S)$. Let $d \in \mathcal{R}$ be chosen such that d enumerates $\mathtt{dom}(S)$ without repetitions. Furthermore, for $i \in \mathbb{N}$ let n be the length of the tuple enumerated by $d(i)$. We set $\psi(i, x) = \varphi_{S(d(i))}(x)$ and $g(i) = n$. Definition 5 directly implies that Conditions (1) and (2) are satisfied.

Sufficiency. First we describe the basic idea for a strategy. Suppose $f \in \mathcal{U}$ and we have already found a program i such that $\psi_i =_{g(i)} f$. Then Condition (2) allows to check whether or not $f(x) = \psi_i(x)$ for all x provided the strategy knows $f(x)$. So, if $f = \psi_i$, the strategy will converge. Otherwise it will find a witness proving $f \neq \psi_i$ and it can restart its search. So, the main problem is to verify $\psi_i =_{g(i)} f$. For overcoming it, let $c \in \mathcal{R}$ be such that $\psi_i = \varphi_{c(i)}$ for all $i \in \mathbb{N}$. Now the idea is to use the input length to provide a bound on $\Phi_{c(i)}(x)$.

The desired strategy S is formally defined as follows. Let z be any fixed number such that $\psi_z \in \mathcal{R}$.

$S(f^n) =$ "Compute $M = \{i \mid i \leqslant n,\ g(i) \leqslant n,\ \Phi_{c(i)}(x) \leqslant n$ and $\psi_i(x) = f(x)$ for all $x \leqslant g(i)\}$. Execute Instruction (I).

(I) If $M = \emptyset$ then output z.
If $M \neq \emptyset$ then let $i = \min M$ and compute $\psi_i(x)$ for all x such that $g(i) < x \leqslant n$. If one of these values is not defined, then $S(f^n)$ is not defined, either.
Otherwise check whether or not $\psi_i =_n f$. If this is the case, output i.
In case $\psi_i \neq_n f$ execute (I) for $M := M \setminus \{i\}$."

It remains to show that $\mathcal{U} \in \mathcal{TOTAL}_\psi(S)$. Let $f \in \mathcal{U}$. If $M = \emptyset$ then we have $\psi_{S(f^n)} = \psi_z \in \mathcal{R}$. If $M \neq \emptyset$, then the definition of M ensures that we already know $\psi_i =_{g(i)} f$. Hence by Condition (2) we also have $\psi_i \in \mathcal{R}$. Thus, $S(f^n)$ is defined and $\psi_{S(f^n)} \in \mathcal{R}$ for all $n \in \mathbb{N}$. Finally, the definition of S directly implies that $(S(f^n))_{n\in\mathbb{N}}$ converges to the least number i with $\psi_i = f$. ∎

We finish this section by showing that $\mathcal{U}_{(\varphi,\Phi)} \notin \mathcal{TOTAL}$ provided the complexity measure (φ, Φ) fulfills a certain intuitive property.

A complexity measure (φ, Φ) is said to satisfy Property `ext` provided for all $i, n \in \mathbb{N}$ such that $\Phi_i(0)\downarrow, \dots, \Phi_i(n)\downarrow$ there is a $\Phi_z \in \mathcal{R}$ such that $\Phi_i =_n \Phi_z$.

Note that the following proof uses an idea from Case and Smith [27].

Theorem 11. $\mathcal{U}_{(\varphi,\Phi)} \notin \mathcal{TOTAL}$ *for all complexity measures* (φ, Φ) *fulfilling Property* `ext`.

Proof. Let $r \in \mathcal{R}$ be chosen such that $\Phi_i = \varphi_{r(i)}$ for all $i \in \mathbb{N}$. Furthermore, by the padding lemma r can be chosen in a way such that r is strongly monotonously increasing, i.e., $r(i) < r(i+1)$ for all $i \in \mathbb{N}$ (cf. Smith [88]). Hence, $\mathtt{Val}(r)$ is recursive. Next, choose $s \in \mathcal{R}$ such that

$$\varphi_{s(j)}(0) = \begin{cases} 0, & \text{if there is an } i \text{ such that } r(i) = j \\ \uparrow, & \text{otherwise .} \end{cases}$$

In order to define $\varphi_{s(j)}$ for all j and all $x > 0$, suppose there is a strategy $S \in \mathcal{P}$ such that $\mathcal{U}_{(\varphi,\Phi)} \in \mathcal{TOTAL}_\varphi(S)$. For all $x \geqslant 0$ let

$$\varphi_{s(j)}(x+1) = \begin{cases} 0, & \text{if } \varphi_j(y)\downarrow, \text{ for all } y \leqslant x \text{ and } S(\varphi_j^x)\downarrow \text{ and} \\ & \varphi_{k_x}(x+1)\downarrow < \varphi_j(x+1), \text{ where } k_x = S(\varphi_j^x) \\ \uparrow, & \text{otherwise .} \end{cases}$$

By the fixed point theorem (cf. Smith [88]) there is a number i such that $\varphi_{s(r(i))} = \varphi_i$. We continue to show inductively that $\varphi_i \in \mathcal{R}$ and that S fails to totally learn Φ_i.

For the induction base, by construction, $\varphi_{s(r(i))}(0) = 0$, since $j = r(i)$. Hence, $\varphi_i(0) = 0$ and thus $\Phi_i(0) = \varphi_{r(i)}(0)\downarrow$.

Next, consider the definition of $\varphi_i(1)$.

$$\varphi_i(1) = \varphi_{s(r(i))}(1) = \begin{cases} 0, & \text{if } \Phi_i(0)\downarrow \text{ and } S(\Phi_i^0)\downarrow \\ & \text{and } \varphi_{k_0}(1)\downarrow < \Phi_i(1), \text{ where } k_0 = S(\Phi_i^0) \\ \uparrow, & \text{otherwise .} \end{cases}$$

Since $\Phi_i(0)\downarrow$ we know by Property $\mathtt{ext}$ that there is a $\Phi_z \in \mathcal{R}$ such that $\Phi_i(0) = \Phi_z(0)$. Consequently, $S(\Phi_i^0)\downarrow$ and $\varphi_{k_0} \in \mathcal{R}$, where $k_0 = S(\Phi_i^0)$. Thus, by Property (2) of the definition of complexity measure, one can effectively decide whether or not $\varphi_{k_0}(1) < \Phi_i(1)$. Clearly, if $\varphi_{k_0}(1) < \Phi_i(1)$, then $\varphi_i(1) = 0$ and hence defined. On the other hand, if $\varphi_{k_0}(1) \geqslant \Phi_i(1)$ then $\Phi_i(1)\downarrow$, too, but, by construction, $\varphi_i(1)\uparrow$, a contradiction to Property (1) of the definition of complexity measure. Hence $\varphi_i(1)$ is defined.

The induction step is done analogously. That is,

$$\varphi_i(x+1) = \varphi_{s(r(i))}(x+1) = \begin{cases} 0, \text{ if } \Phi_i(y)\downarrow, \text{ for all } y \leqslant x \text{ and } S(\Phi_i^x)\downarrow \text{ and} \\ \quad \varphi_{k_x}(x+1)\downarrow < \Phi_i(x+1), \text{ where } k_x = S(\Phi_i^x) \\ \uparrow, \text{ otherwise .} \end{cases}$$

By the induction hypothesis, $\Phi_i(y)\downarrow$ for all $y \leqslant x$ and thus, by Property $\mathtt{ext}$, there is a $\Phi_z \in \mathcal{R}$ such that $\Phi_i =_x \Phi_z$ and therefore $S(\Phi_i^x)\downarrow$. Let $k_x = S(\Phi_i^x)$, then $\varphi_{k_x} \in \mathcal{R}$ and one can effectively decide whether or not $\varphi_{k_x}(x+1) < \Phi_i(x+1)$. If it is, $\varphi_i(x+1) = 0$ and thus $\Phi_i(x+1)\downarrow$. If it is not, we have $\varphi_{k_x}(x+1) \geqslant \Phi_i(x+1)$ but $\varphi_i(x+1)\uparrow$, a contradiction to Property (1) of the definition of complexity measure. Hence, $\varphi_i(x+1)$ is defined.

Therefore, we obtain $\varphi_i \in \mathcal{R}$ and hence $\Phi_i \in \mathcal{R}$, too. Consequently, $\Phi_i \in \mathcal{U}_{(\varphi,\Phi)}$. By supposition, S has to learn Φ_i, i.e., the sequence $(k_x)_{x\in\mathbb{N}}$ has to converge, say to k, and k must be a φ-program for Φ_i. But by construction we have $\varphi_k(x+1) < \Phi_i(x+1)$ for all but finitely many $x \in \mathbb{N}$, a contradiction. ∎

Now we are ready to explore the other ways mentioned above to enlarge the learning capabilities of $\mathcal{R}\text{-}\mathcal{TOTAL}$. This brings us directly to another subject Rolf Wiehagen has been interested in for many years, i.e., learning and consistency.

4. Learning and Consistency – Part I

Looking back at the proof of Theorem 2, we see that an $\mathcal{R}$-total strategy is always completely and correctly reflecting the data seen so far. Such a hypothesis is called *consistent*. Hypotheses not behaving thus are said to be *inconsistent*. Consequently, if a strategy has already seen the examples $(x_0, f(x_0)), \dots, (x_n, f(x_n))$ and is hypothesizing the function g and if g is inconsistent, then there must be a $k \leqslant n$ such that $g(x_k) \neq f(x_k)$. Note that there are two possible reasons for g to differ from f on argument x_k. Either $g(x_k)\uparrow$ or $g(x_k)\downarrow$ but does not equal $f(x_k)$. In any way, an inconsistent hypothesis is not only wrong but it is wrong on an argument for which the learning strategy does already know the correct value. Thus, one may be tempted to completely exclude strategies producing inconsistent hypotheses. So, let us follow this temptation and let us see what we get. We start with the strongest version of consistent learning which has already been considered in [19]. Note that Blum and Blum [19] called this form of consistency the *overkill property*.

Definition 6 (Blum and Blum [19]). *Let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\psi \in \mathcal{P}^2$.
$\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}_{\psi}^{arb}$ *if there is a strategy* $S \in \mathcal{R}$ *such that*

(1) *for all* $f \in \mathcal{U}$ *and every* $X \in \Pi(\mathbb{N})$, *there is a* $j \in \mathbb{N}$ *such that* $\psi_j = f$, *and* $(S(f^{X,n}))_{n\in\mathbb{N}}$ *converges to* j,

(2) $\psi_{S(f^{X,n})}(x_m) = f(x_m)$ *for every permutation* $X \in \Pi(\mathbb{N})$, $f \in \mathcal{R}$, $n \in \mathbb{N}$, *and* $m \leqslant n$.

$\mathcal{T}\text{-}\mathcal{CONS}_{\psi}^{arb}(S)$ *as well as* $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$ *are defined in analogy to the above.*

That means a $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$ strategy is required to return consistent hypotheses even if the input does not belong to any function in the target class $\mathcal{U}$.

Next, we introduce a new family of complexity classes. Let $h \in \mathcal{R}^2$; then

$$\mathcal{R}_h = \{\varphi_i \mid \forall^\infty n[\Phi_i(n) \leqslant h(n, \varphi_i(n))]\} \cap \mathcal{R}$$

is called *honesty complexity class*. So, honesty means that every function $f \in \mathcal{R}_h$ does possess a φ-program i computing it, i.e., $\varphi_i = f$ and the complexity of this φ-program can be bounded by using the function $h \in \mathcal{R}^2$ *and* the function values $f(n)$.

Furthermore, we introduce a new family of numberings.

Definition 7. *A numbering* $\psi \in \mathcal{P}^2$ *is said to be* measurable *if the predicate "*$\psi_i(x) = y$*" is uniformly recursive in* i, x, y.

The next theorem completely characterizes $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$ in terms of complexity and of computable numberings. The proof presented below is a combination of results from Blum and Blum [19] (Assertion (1) and (2)) and from McCreight and Meyer [74] who showed the equivalence of Assertion (2) and (3).

Theorem 12 (Blum and Blum [19], McCreight and Meyer [74]).

Let (φ, Φ) *be any complexity measure and let* $\mathcal{U} \subseteq \mathcal{R}$*. Then the following conditions are equivalent.*

(1) $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}^{arb}$.

(2) *There is a function* $h \in \mathcal{R}^2$ *such that* $\mathcal{U} \subseteq \mathcal{R}_h$.

(3) *There is a measurable numbering* $\psi \in \mathcal{P}^2$ *such that* $\mathcal{U} \subseteq \mathcal{P}_\psi$.

Proof. The proof is done by showing three claims.

Claim 1. (1) *implies* (2).

Let $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}^{arb}_{\varphi}(S)$ be witnessed by $S \in \mathcal{R}$ and $\varphi \in \mathsf{G\ddot{o}d}$. Furthermore, let $c_2 \colon \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$ be the standard Cantor coding of all pairs of natural numbers. We define an order $\prec$ on $\mathbb{N} \times \mathbb{N}$. Let (x_1, y_1), $(x_2, y_2) \in \mathbb{N} \times \mathbb{N}$. Then

$$(x_1, y_1) \prec (x_2, y_2) \text{ if and only if } c_2(x_1, y_1) < c_2(x_2, y_2) \ .$$

Clearly, $\prec$ is computable.

Furthermore, for (x, y) we denote by $SEQ(x, y)$ the set of all finite sequences $\sigma = ((x_0, y_0), \ldots, (x_n, y_n), (x, y))$ for which $(x_0, y_0) \prec \cdots \prec (x_n, y_n) \prec (x, y)$. Note that for every pair (x, y) the set $SEQ(x, y)$ is finite and computable. Since S is consistent in the sense of $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$ we additionally have

$$\varphi_{S(\langle\sigma\rangle)}(x) = y \quad \text{for all } \sigma \in SEQ(x, y) \ . \tag{1}$$

Now we are ready to define the desired function h. For all $x, y \in \mathbb{N}$ let

$$h(x, y) = \max\{\Phi_{S(\langle\sigma\rangle)}(x) \mid \sigma \in SEQ(x, y)\} \ .$$

Since for every pair (x, y) the set $SEQ(x, y)$ is finite and computable, by (1) we directly get $h \in \mathcal{R}^2$.

Now let $f \in \mathcal{U}$. We have to show $f \in \mathcal{R}_h$. Note that $\prec$ induces precisely one enumeration $(x_0, f(x_0))$, $(x_1, f(x_1)), \ldots$ of the graph of f. By the definition of $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$ the strategy S has to converge to a number j with $\varphi_j = f$ when successively fed this enumeration. Thus, for all sufficiently large n we have

$$S(\langle(x_0, f(x_0)), \ldots, (x_n, f(x_n))\rangle) = j \ .$$

By the definition of h we can directly conclude $\Phi_j(x_n) \leqslant h(x_n, \varphi_j(x_n))$ for all sufficiently large n. Consequently, $f \in \mathcal{R}_h$, and Claim 1 is shown.

Claim 2. (2) *implies* (3).

Let $h \in \mathcal{R}^2$ and let $f \in \mathcal{R}_h$. Then there exists a triple (j, n, k) such that $\varphi_j = f$, $\Phi_j(x) \leqslant k$ for all $x \leqslant n$ and $\Phi_j(x) \leqslant h(x, \varphi_j(x))$ for all $x > n$. Using ideas similar to those applied in the proof of the sufficiency part of Theorem 4 we can define the desired numbering ψ. Again, let c_3 be the canonical enumeration of $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$. For $c_3(i) = (j, n, k)$ and $x \in \mathbb{N}$ we define

$$\psi(i,x) = \begin{cases} y, & \text{if } [x \leqslant n \rightarrow \Phi_j(x) \leqslant k \text{ and } \varphi_j(x) = y] \\ & \text{or } [x > n \rightarrow \Phi_j(x) \leqslant h(x,y) \text{ and } \varphi_j(x) = y] \\ \uparrow, & \text{otherwise.} \end{cases}$$

Obviously, we have $\psi \in \mathcal{P}^2$ and by the observation made above it is easy to see that $\mathcal{U} \subseteq \mathcal{P}_\psi$. It remains to show that ψ is measurable. So, we have to provide an algorithm uniformly deciding on input i, x, y whether or not $\psi(i, x) = y$. The desired algorithm is displayed in Figure 1. Note that rounded rectangles denote tests. This proves Claim 2.

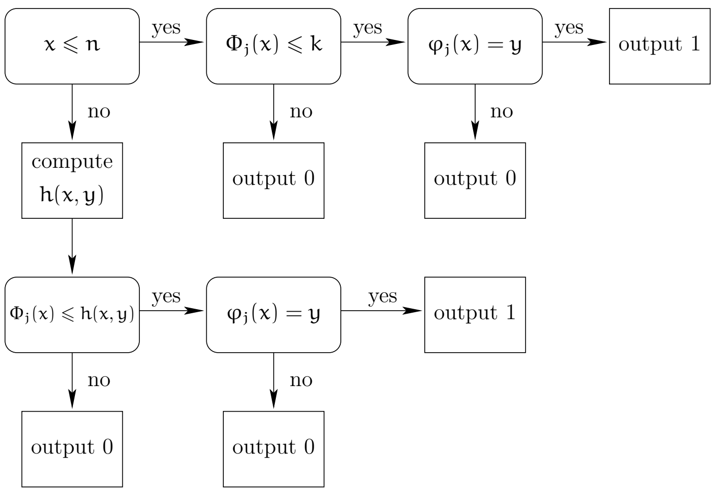

Figure 1: An algorithm uniformly deciding on input i, x, y whether or not $\psi(i, x) = y$; here $(j, n, k) = c_3(i)$.

Claim 3. (3) *implies* (1).

Let $\mathcal{U} \subseteq \mathcal{R}$ and let $\psi \in \mathcal{P}^2$ be a measurable numbering such that $\mathcal{U} \subseteq \mathcal{R}_\psi$. Moreover, as in the proof of Theorem 2 we chose $\chi \in \mathcal{R}^2$ such that $\mathcal{U}_0 = \mathcal{R}_\chi$. Again, we set $\tau_{2i} = \psi_i$ and $\tau_{2i+1} = \chi_i$ for all $i \in \mathbb{N}$. Obviously, $\tau \in \mathcal{P}^2$, τ is measurable, and $\mathcal{U} \subseteq \mathcal{R}_\tau$. Now let $X \in \Pi(\mathbb{N})$ and $n \in \mathbb{N}$. We define

$S(f^{X,n}) =$ "Search the least i such that $\tau_i(x_m) = f(x_m)$ for all $0 \leqslant m \leqslant n$. If such an i has been found, output i."

Since τ is measurable, it is easy to see that $S \in \mathcal{R}$. Moreover, if $f \in \mathcal{U}$, then the sequence $(S(f^{X,n}))_{n \in \mathbb{N}}$ has to converge, since the search can never go beyond the least τ-program j with $\tau_j = f$. When converging, say to j, the strategy yields $\tau_j = f$. Thus, $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}_\tau^{arb}(S)$.

This proves Claim 3, and hence the theorem is shown. ∎

This theorem directly allows the following corollary.

Corollary 13 (Blum and Blum [19]).

(1) $\mathcal{R}\text{-}\mathcal{TOTAL} \subset \mathcal{T}\text{-}\mathcal{CONS}^{arb}$ *and*

(2) $\mathcal{R}\text{-}\mathcal{TOTAL} \cap \wp(\mathcal{R}_{\{0,1\}}) = \mathcal{T}\text{-}\mathcal{CONS}^{arb} \cap \wp(\mathcal{R}_{\{0,1\}})$.

(3) $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$ *is closed under recursively enumerable union.*

Proof. For the first part, by Lemma 1 and Theorem 2 we have $\mathcal{U}_{(\varphi,\Phi)} \notin \mathcal{R}\text{-}\mathcal{TOTAL}$. On the other hand, for every complexity measure (φ, Φ), $\Phi \in \mathcal{P}^2$ is measurable. Hence $\mathcal{U}_{(\varphi,\Phi)} \in \mathcal{T}\text{-}\mathcal{CONS}^{arb}$. Consequently, $\mathcal{T}\text{-}\mathcal{CONS}^{arb} \setminus \mathcal{R}\text{-}\mathcal{TOTAL} \neq \emptyset$. Furthermore, $\mathcal{R}\text{-}\mathcal{TOTAL} \subseteq \mathcal{T}\text{-}\mathcal{CONS}^{arb}$ by Theorem 2 and Theorem 12.

For the second part, if $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}^{arb}$ then there is a function $h \in \mathcal{R}^2$ such that $\mathcal{U} \subseteq \mathcal{R}_h$. But since $\mathcal{U} \subseteq \mathcal{R}_{\{0,1\}}$, we can define $t(x) = h(x, 0) + h(x, 1)$ for all $x \in \mathbb{N}$. Hence, we get $\mathcal{U} \subseteq \mathcal{R}_t$, and thus by Theorem 2 we know $\mathcal{U} \in \mathcal{NUM} = \mathcal{R}\text{-}\mathcal{TOTAL}$.

Assertion (3) is proved by using Theorem 12. Let $(S_i)_{i \in \mathbb{N}}$ be a recursive enumeration of strategies fulfilling the requirements of $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$. Without loss of generality we can assume that all strategies S_i learn with respect to some fixed Gödel numbering φ. As the proof of Claim 1 in the demonstration of Theorem 12 shows, for every strategy S_i we can effectively obtain a function $h_i \in \mathcal{R}^2$ such that $\mathcal{T}\text{-}\mathcal{CONS}^{arb}_{\varphi}(S_i) \subseteq \wp(\mathcal{R}_{h_i})$. We define

$$h(x, y) = \max\{h_i(x, y) \mid i \leqslant x\} \text{ for all } x, y \in \mathbb{N} \ .$$

Clearly, $h \in \mathcal{R}^2$ and by construction $\bigcup_{i \in \mathbb{N}} \mathcal{R}_{h_i} \subseteq \mathcal{R}_h$. Applying again Theorem 12 we get that there is a strategy S such that $\mathcal{R}_h \in \mathcal{T}\text{-}\mathcal{CONS}^{arb}_{\varphi}(S)$. Consequently, $\bigcup_{i \in \mathbb{N}} \mathcal{T}\text{-}\mathcal{CONS}^{arb}_{\varphi}(S_i) \subseteq \mathcal{T}\text{-}\mathcal{CONS}^{arb}_{\varphi}(S)$. ∎

Furthermore, $\mathcal{TOTAL}$ and $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$ both extend the learning capabilities of $\mathcal{R}\text{-}\mathcal{TOTAL}$ but in different directions. Before showing this, we consider the variant of $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$ where the strategy is only required to learn from input presented in natural order. The resulting learning type is denoted by $\mathcal{T}\text{-}\mathcal{CONS}$.

Corollary 14. $\mathcal{TOTAL} \# \mathcal{T}\text{-}\mathcal{CONS}^{arb}$

Proof. By Theorem 11 we have $\mathcal{U}_{(\varphi,\Phi)} \in \mathcal{T}\text{-}\mathcal{CONS}^{arb} \setminus \mathcal{TOTAL}$. On the other hand, $\mathcal{U}_{sd} \in \mathcal{TOTAL}$. We claim that $\mathcal{U}_{sd} \notin \mathcal{T}\text{-}\mathcal{CONS}$, and thus we also have $\mathcal{U}_{sd} \notin \mathcal{T}\text{-}\mathcal{CONS}^{arb}$.

Let φ be any fixed Gödel numbering. Suppose there is a strategy $S \in \mathcal{R}$ such that $\mathcal{U}_{sd} \in \mathcal{T}\text{-}\mathcal{CONS}_{\varphi}(S)$. By an easy application of the fixed point theorem we can construct a function f such that $f = \varphi_i$, $f(0) = i$ and for all $n \in \mathbb{N}$

$$f(n+1) = \begin{cases} 0, & \text{if } S(f^n 0) \neq S(f^n) \\ 1, & \text{if } S(f^n 0) = S(f^n) \text{ and } S(f^n 1) \neq S(f^n). \end{cases}$$

Note that one of the two cases in the definition of f must happen for all $n \geqslant 1$. Thus, we clearly have $f \in \mathcal{U}_{sd}$. On the other hand, $S(f^n) \neq S(f^{n+1})$ for all $n \in \mathbb{N}$ a contradiction to $\mathcal{U}_{sd} \in \mathcal{T}\text{-}\mathcal{CONS}_{\varphi}(S)$. Hence $\mathcal{U}_{sd} \notin \mathcal{T}\text{-}\mathcal{CONS}$. ∎

Note that the proof of this Corollary also showed that for every $\mathcal{T}$-consistent strategy $S \in \mathcal{R}$ one can effectively construct a function f such that $\{f\} \notin \mathcal{T}\text{-}\mathcal{CONS}_{\varphi}(S)$.

We finish this section by mentioning that for $\mathcal{T}$-consistent learning identification from arbitrarily ordered input and learning from input presented in natural order makes a difference. Thus, the following theorem nicely contrasts with Theorems 1 and 9. For a proof we refer the reader to Grieser [50].

Theorem 15. $\mathcal{T}\text{-}\mathcal{CONS}^{arb} \subset \mathcal{T}\text{-}\mathcal{CONS}$

We continue by defining some more concepts of learning. This is done in the next section.

5. Defining More Learning Models

So far, we have started from a learning model which, at first glance looked quite natural, i.e., $\mathcal{R}\text{-}\mathcal{TOTAL}^{arb}$ and continued by looking for possibilities to enlarge its learning power. Though, conceptually, we shall follow this line of presentation, it is technically advantageous to introduce several new concepts of learning at once in this section.

The following learning model is the one with which it all started, i.e., Gold's famous *learning in the limit* model. In this model, all requirements on the intermediate hypotheses such as being ψ-programs of recursive functions or being consistent are dropped.

Definition 8 (Gold [47, 48]). *Let $\mathcal{U} \subseteq \mathcal{R}$ and let $\psi \in \mathcal{P}^2$. The class $\mathcal{U}$ is said to be* learnable in the limit with respect to ψ *if there is a strategy $S \in \mathcal{P}$ such that for each function $f \in \mathcal{U}$,*

(1) *for all $n \in \mathbb{N}$, $S(f^n)$ is defined,*

(2) *there is a $j \in \mathbb{N}$ such that $\psi_j = f$ and the sequence $(S(f^n))_{n\in\mathbb{N}}$ converges to j.*

If $\mathcal{U}$ is learnable in the limit with respect to ψ by a strategy S, we write $\mathcal{U} \in \mathcal{LIM}_{\psi}(S)$. Let $\mathcal{LIM}_{\psi} = \{\mathcal{U} \mid \mathcal{U}$ is learnable in the limit w.r.t. $\psi\}$, and let $\mathcal{LIM} = \bigcup_{\psi\in\mathcal{P}^2} \mathcal{LIM}_{\psi}$.

Again, some remarks are mandatory here. Note that $\mathcal{LIM}_{\varphi} = \mathcal{LIM}$ for any Gödel numbering φ. This can be shown by using exactly the same ideas as above (cf. Lemma 2). In the above definition $\mathcal{LIM}$ stands for "limit." There are also other notations around to denote the learning type $\mathcal{LIM}$. For example, in [11, 12, 13] the notation GN is used. Here GN stands for Gödel numbering. Case and Smith [27] coined the notation EX which stands for "explain."

As we have seen above when studying the learning types $\mathcal{R}\text{-}\mathcal{TOTAL}$ and $\mathcal{TOTAL}$, it can make a difference with respect to the resulting learning power whether or not we require the strategy to be in $\mathcal{R}$ or in $\mathcal{P}$ (cf. Theorem 7). On the other hand, the learning type $\mathcal{LIM}$ is invariant to the demand $S \in \mathcal{R}$ instead of $S \in \mathcal{P}$. This has already been shown by Gold [47] and for the sake of completeness we include this result here.

Theorem 16 (Gold [47]). *Let* (φ, Φ) *be a complexity measure. There is a function* $s \in \mathcal{R}$ *such that* $\varphi_{s(i)} \in \mathcal{R}$ *and* $\mathcal{LIM}_\varphi(\varphi_i) \subseteq \mathcal{LIM}_\varphi(\varphi_{s(i)})$ *for all* $i \in \mathbb{N}$.

Proof. For every $(y_0, \ldots, y_n) \in \mathbb{N}^*$ we set

$$\varphi_{s(i)}(\langle(y_0, \ldots, y_n)\rangle) = \begin{cases} 0, & \text{if } \Phi_i(\langle(y_0, \ldots, y_x)\rangle) > n \\ & \text{for all } x \leqslant n \\ \varphi_i(\langle(y_0, \ldots, y_{x'})\rangle), & \text{if } x' \text{ is the biggest } x \leqslant n \text{ such} \\ & \text{that } \Phi_i(\langle(y_0, \ldots, y_x)\rangle) \leqslant n\ . \end{cases}$$

Now let $f \in \mathcal{R}$ be such that $(\varphi_i(f^n))_{n\in\mathbb{N}}$ converges, say to j, and $\varphi_j = f$. Then, by construction, the sequence $(\varphi_{s(i)}(f^n))_{n\in\mathbb{N}}$ also converges to j, but possibly with a certain delay. Thus, $\varphi_{s(i)}$ learns f in the limit, too. ∎

Hence, there exists a numbering $\psi \in \mathcal{R}^2$ such that for every $\mathcal{U} \in \mathcal{LIM}$ there is a strategy $S \in \mathcal{R}_\psi$ satisfying $\mathcal{U} \in \mathcal{LIM}(S)$. Clearly, it suffices to set $\psi_i = \varphi_{s(i)}$. This in turn implies that there is no effective procedure to construct for every strategy $\varphi_{s(i)}$ a function $f_i \in \mathcal{R}$ such that $\{f_i\} \notin \mathcal{LIM}(\varphi_{s(i)})$. In order to see this, suppose the converse. Hence, the class $\{f_i \mid i \in \mathbb{N}\}$ would be in $\mathcal{NUM} \setminus \mathcal{LIM}$, a contradiction, since we obviously have $\mathcal{NUM} \subseteq \mathcal{LIM}$.

Furthermore, a straightforward modification of Definition 8 yields $\mathcal{LIM}^{arb}$, i.e., learning in the limit from arbitrary input. Using the same idea as in the proof of Theorem 1 one can easily show that $\mathcal{LIM} = \mathcal{LIM}^{arb}$.

5.1. *Varying the Mode of Convergence*

Note that in general it is not decidable whether or not a strategy has already converged when successively fed some graph of a function. With the next definition we consider a special case where it has to be decidable whether or not a strategy has already learned its input function. That is, we replace the requirement that the sequence of all created hypotheses "has to *converge*" by "has to *converge finitely*."

Definition 9 (Gold [48], Trakhtenbrot and Barzdin [96]). *Let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\psi \in \mathcal{P}^2$. *The class* $\mathcal{U}$ *is said to be* finitely learnable with respect to ψ *if there is a strategy* $S \in \mathcal{P}$ *such that for any function* $f \in \mathcal{U}$,

(1) *for all* $n \in \mathbb{N}$, $S(f^n)$ *is defined,*

(2) *there is a* $j \in \mathbb{N}$ *such that* $\psi_j = f$ *and the sequence* $(S(f^n))_{n\in\mathbb{N}}$ *finitely converges to* j.

If $\mathcal{U}$ *is finitely learnable with respect to* ψ *by a strategy* S, *we write* $\mathcal{U} \in \mathcal{FIN}_\psi(S)$. *Let* $\mathcal{FIN}_\psi = \{\mathcal{U} \mid \mathcal{U}$ is finitely learnable w.r.t. $\psi\}$, *and let* $\mathcal{FIN} = \bigcup_{\psi\in\mathcal{P}^2} \mathcal{FIN}_\psi$.

Though the following result is not hard to prove, it provides some nice insight into the limitations of finite learning. For stating it, we need the notion of accumulation point. Let $\mathcal{U} \subseteq \mathcal{R}$; then a function $f \in \mathcal{R}$ is said to be an *accumulation point* of $\mathcal{U}$ if for every $n \in \mathbb{N}$ there is a function $\hat{f} \in \mathcal{U}$ such that $f =_n \hat{f}$ but $f \neq \hat{f}$.

Theorem 17 (Lindner [71]). *Let $\mathcal{U} \subseteq \mathcal{R}$ be any class such that $\mathcal{U} \in \mathcal{FIN}$. Then $\mathcal{U}$ cannot contain any accumulation point.*

Proof. Suppose the converse, i.e., there is a class $\mathcal{U} \in \mathcal{FIN}$ containing an accumulation point f. Let $S \in \mathcal{P}$ such that $\mathcal{U} \in \mathcal{FIN}(S)$. Then there must exist an $n \in \mathbb{N}$ such $S(f^n) = S(f^{n+1}) = j$. That is, the sequence $(S(f^n))_{n \in \mathbb{N}}$ has finitely converged to j and $\varphi_j = f$ must hold. On the other hand, since f is an accumulation point, there must be an $\hat{f} \in \mathcal{U}$ such that $f =_{n+1} \hat{f}$ but $f \neq \hat{f}$. Clearly, by the definition of finite convergence we have $S(\hat{f}^n) = S(\hat{f}^{n+1}) = j$, too, but $\varphi_j = f \neq \hat{f}$. This is a contradiction to $\mathcal{U} \in \mathcal{FIN}(S)$. ∎

This theorem directly yields the following corollary.

Corollary 18. $\mathcal{R}\text{-}\mathcal{TOTAL} \# \mathcal{FIN}$

Proof. $\mathcal{FIN} \setminus \mathcal{R}\text{-}\mathcal{TOTAL} \neq \emptyset$ is witnessed by $\mathcal{U}_{sd}$. Moreover, the function $0^\infty \in \mathcal{U}_0$ is clearly an accumulation point of the class $\mathcal{U}_0$. Thus, by Theorems 17 and 2 we get $\mathcal{U}_0 \in \mathcal{R}\text{-}\mathcal{TOTAL} \setminus \mathcal{FIN}$. ∎

Note that Theorem 50 provides a complete answer to the question under which circumstances a class $\mathcal{U} \subseteq \mathcal{R}$ is finitely learnable.

Next, we look at another mode of convergence which goes back to Feldman [31] who called it *matching in the limit* and considered it in the setting of learning languages. The difference to the mode of convergence used in Definition 8, which is actually *syntactic convergence*, is to relax the requirement that the sequence of hypotheses has to converge to a correct program by *semantic convergence*. Here by semantic convergence we mean that after some point *all hypotheses are correct* but not necessarily identical. Nowadays, the resulting learning model is usually referred to as *behaviorally correct* learning. This term has been coined by Case and Smith [27]. As far as learning of recursive functions is concerned, behaviorally correct learning was formalized by Barzdin [9, 16].

Definition 10 (Barzdin [9, 16]). *Let $\mathcal{U} \subseteq \mathcal{R}$ and let $\psi \in \mathcal{P}^2$. The class $\mathcal{U}$ is said to be* behaviorally correctly learnable with respect to ψ *if there is a strategy $S \in \mathcal{P}$ such that for each function $f \in \mathcal{U}$,*

(1) *for all $n \in \mathbb{N}$, $S(f^n)$ is defined,*

(2) $\psi_{S(f^n)} = f$ *for all but finitely many $n \in \mathbb{N}$.*

If $\mathcal{U}$ is behaviorally correctly learnable with respect to ψ by a strategy S, we write $\mathcal{U} \in \mathcal{BC}_\psi(S)$. $\mathcal{BC}_\psi$ and $\mathcal{BC}$ are defined analogously to the above above.

Clearly, we have $\mathcal{LIM} \subseteq \mathcal{BC}$. On the other hand, even $\mathcal{BC}$ learning is not trivial, i.e., we have $\mathcal{R} \notin \mathcal{BC}$. This is a direct consequence of the next theorem which shows the even stronger result that $\mathcal{BC}$ is not closed under union. In the proof below we use the convention that 0^k denotes the empty string for $k = 0$. When we identify a function with the sequence of its values then we mean by $i0^0 20^\infty$ the function f expressed by $i20^\infty$, i.e., $f(0) = i$, $f(1) = 2$ and $f(x) = 0$ for all $x \geqslant 2$.

Theorem 19 (Barzdin [16]). $\mathcal{BC}$ *is not closed under finite union.*

Proof. For showing the theorem it suffices to prove that $\mathcal{U}_{sd} \cup \mathcal{U}_0 \notin \mathcal{BC}$. The proof is done indirectly. Suppose the converse, i.e., there is a strategy $S \in \mathcal{P}$ such that $\mathcal{U}_{sd} \cup \mathcal{U}_0 \in \mathcal{BC}(S)$. Then we can directly conclude $S \in \mathcal{R}$.

Now we have to fool the strategy S such that it would have to "change its mind semantically" infinitely often in order to learn the function to be constructed. For all $i \in \mathbb{N}$ we define a function f_i as follows. Set $f_i(0) = i$ for all $i \in \mathbb{N}$. The definition continues in stages.

Stage 1. Try to compute $\varphi_{S(\langle i \rangle)}(1)$, $\varphi_{S(\langle i0 \rangle)}(2)$, $\dots$, $\varphi_{S(\langle i0^k \rangle)}(k+1)$, $\dots$, until the first value k_1 is found such that $\varphi_{S(\langle i0^{k_1} \rangle)}(k_1+1)\downarrow$.

Let $y_1 = \varphi_{S(\langle i0^{k_1} \rangle)}(k_1+1)$. Then we set $f_i(x) = 0$ for all $1 \leqslant x \leqslant k_1$ and $f_i(k_1+1) = y_1 + 1$.
Goto Stage 2.

If none of the values $\varphi_{S(\langle i \rangle)}(1)$, $\varphi_{S(\langle i0^k \rangle)}(k+1)$, $k \in \mathbb{N}^+$, is defined, then Stage 1 is not left. But in this case we are already done, since then $\{i0^\infty\} \notin \mathcal{BC}(S)$.

For making the proof easier to access, we also include Stage 2 here.

Stage 2. Try to compute $\varphi_{S(\langle i0^{k_1} f(k_1+1) \rangle)}(k_1+2)$, $\varphi_{S(\langle i0^{k_1} f(k_1+1)0 \rangle)}(k_1+3)$, $\dots$, $\varphi_{S(\langle i0^{k_1} f(k_1+1)0^k \rangle)}(k_1+k+2)$, $\dots$, until the first value k_2 is found such that $\varphi_{S(\langle i0^{k_1} f(k_1+1)0^{k_2} \rangle)}(k_1+k_2+2)\downarrow$.

Let $y_2 = \varphi_{S(\langle i0^{k_1} f(k_1+1)0^{k_2} \rangle)}(k_1+k_2+2)$. Then we set $f_i(x) = 0$ for all $k_1+2 \leqslant x \leqslant k_1+k_2+1$ and $f_i(k_1+k_2+2) = y_2+1$.
Goto Stage 3.

Again, if none of the values $\varphi_{S(\langle i0^{k_1} f(k_1+1) \rangle)}(k_1+2)$, $\varphi_{S(\langle i0^{k_1} f(k_1+1)0^k \rangle)}(k_1+k+2)$, $k \in \mathbb{N}^+$, is defined, then Stage 2 is not left. But in this case we are again done, since then $\{i0^{k_1} f(k_1+1)0^\infty\} \notin \mathcal{BC}(S)$.

Now this construction is iterated. We assume that Stage n, $n > 1$ has been left. Then numbers $k_1, \dots, k_n$ have been found such that

$$\varphi_{S(f_i^{k_1+\cdots+k_\ell+\ell})}(k_1+\cdots+k_\ell+\ell)\downarrow \quad \text{for } \ell = 1, \dots, n\ .$$

So, $f_i(x)$ is already defined for all $0 \leqslant x \leqslant k_1+\cdots+k_n+n$.

Stage $n+1$, $n \geqslant 2$. Try to compute

$$\varphi_{S(\langle i0^{k_1}f_i(k_1+1)\cdots 0^{k_n}f_i(k_1+\cdots+k_n+n)\rangle)}(k_1+\cdots+k_n+n+1)$$
$$\varphi_{S(\langle i0^{k_1}f_i(k_1+1)\cdots 0^{k_n}f_i(k_1+\cdots+k_n+n)0\rangle)}(k_1+\cdots+k_n+n+2)$$
$$\vdots$$
$$\varphi_{S(\langle i0^{k_1}f_i(k_1+1)\cdots 0^{k_n}f_i(k_1+\cdots+k_n+n)0^{k}\rangle)}(k_1+\cdots+k_n+n+k+1)$$
$$\vdots$$

until the first value k_{n+1} is found such that

$$\varphi_{S(\langle i0^{k_1}f_i(k_1+1)\cdots 0^{k_n}f_i(k_1+\cdots+k_n+n)0^{k_{n+1}}\rangle)}(k_1+\cdots+k_n+n+k_{n+1}+1)\downarrow .$$

Let $y_{n+1} = \varphi_{S(\langle i0^{k_1}f_i(k_1+1)\cdots 0^{k_n}f_i(k_1+\cdots+k_n+n)0^{k_{n+1}}\rangle)}(k_1+\cdots+k_n+n+k_{n+1}+1)$. Then we set $f_i(x) = 0$ for all $k_1+\cdots+k_n+n+1 \leqslant x \leqslant k_1+\cdots+k_n+n+k_{n+1}$, and set $f_i(k_1+\cdots+k_n+k_{n+1}+n+1) = y_{n+1}+1$.

As before, if Stage $n+1$ is not left, we are already done. Thus, it remains to consider the case that Stage n is left for all $n \geqslant 1$. Let $s \in \mathcal{R}$ be chosen such that $\varphi_{s(i)} = f_i$ for all $i \in \mathbb{N}$. By the fixed point theorem (cf., e.g., Smith [88]) there is a number j such that $\varphi_{s(j)} = \varphi_j$. Since $f_j = \varphi_{s(j)} = \varphi_j$ and $f_j(0) = j$ we get $f_j \in \mathcal{U}_{sd}$. But by construction we have $\varphi_{S(f_j^{k_1})}(k_1+1) \neq f_j(k_1+1)$, $\varphi_{S(f_j^{k_1+k_2+1})}(k_1+k_2+2) \neq f_j(k_1+k_2+2)$, $\dots$, $\varphi_{S(f_j^{k_1+\cdots k_n+n-1})}(k_1+\cdots+k_n+n) \neq f_j(k_1+\cdots+k_n+n)$, $\dots$. Therefore, when successively fed f_j^n the strategy S outputs infinitely many wrong hypotheses, and thus $f_j \notin \mathcal{BC}(S)$, a contradiction to $\mathcal{U}_0 \cup \mathcal{U}_{sd} \in \mathcal{BC}(S)$. ∎

This proof directly yields the following corollary.

Corollary 20.

(1) $\mathcal{R} \notin \mathcal{BC}$.

(2) $\mathcal{LIM}$ *is not closed under finite union.*

(3) $\mathcal{R} \notin \mathcal{LIM}$.

Proof. (1) is a direct consequence of Theorem 19. Clearly, $\mathcal{U}_{sd}, \mathcal{U}_0 \in \mathcal{LIM}$ and $\mathcal{LIM} \subseteq \mathcal{BC}$. Since $\mathcal{U}_{sd} \cup \mathcal{U}_0 \notin \mathcal{BC}$, Assertion (2) follows. Finally, (3) is directly implied by Assertion (2). ∎

Adleman and Blum [1] have shown that, under canonical formalization, the degree of the algorithmic unsolvability of "$\mathcal{R} \in \mathcal{LIM}$" is strictly less than the degree of the algorithmic unsolvability of the halting problem. Brand [23] studied the related problem of identifying all partial recursive functions. Of course, it is also algorithmically unsolvable but its degree and the degree of the halting problem are equivalent.

Moreover, Apsītis *et al.* [7] investigated the problem in which cases unions of identifiable classes are also necessarily identifiable and obtained many interesting and beautiful results.

On the other hand, many more function classes are learnable behaviorally correctly than are learnable in the limit. In order to state this result and for pointing to another interesting property of behaviorally correct learning, we modify Definition 8 by relaxing the *learning goal.* By $\mathcal{R}_*$ and $\mathfrak{T}_*$ we denote the class of all functions $f \in \mathcal{P}$ and $f \in \mathfrak{P}$, respectively, for which $\mathtt{dom}(f)$ is cofinite. For $f, g \in \mathfrak{T}_*$ and $a \in \mathbb{N}$ we write $f =^a g$ and $f =^* g$ if $|\{x \in \mathbb{N} \mid f(x) \neq g(x)\}| \leqslant a$ and $|\{x \in \mathbb{N} \mid f(x) \neq g(x)\}| < \infty$, respectively. Note that there are three possibilities for a number x to belong to the sets just considered: both $f(x)\downarrow$ and $g(x)\downarrow$, but $f(x) \neq g(x)$, or $f(x)\downarrow$ while $g(x)\uparrow$, or $f(x)\uparrow$ and $g(x)\downarrow$.

Definition 11 (Case and Smith [27]). *Let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\psi \in \mathcal{P}^2$. *Let* $a \in \mathbb{N} \cup \{*\}$. *The class* $\mathcal{U}$ *is said to be* learnable in the limit with a anomalies (in case $a = *$: with finitely many anomalies) with respect to ψ *if there is a strategy* $S \in \mathcal{P}$ *such that for each function* $f \in \mathcal{U}$,

(1) *for all* $n \in \mathbb{N}$, $S(f^n)$ *is defined,*

(2) *there is a* $j \in \mathbb{N}$ *such that* $\psi_j =^a f$ *and the sequence* $(S(f^n))_{n\in\mathbb{N}}$ *converges to* j.

This is denoted by $\mathcal{U} \in \mathcal{LIM}^a_\psi$ *for short. The notions* $\mathcal{LIM}^a_\psi$ *and* $\mathcal{LIM}^a$ *are defined in the usual way.*

Note that for $a = 0$, the inference type $\mathcal{LIM}^0$ coincides with $\mathcal{LIM}$ by definition. The following theorem establishes an infinite hierarchy in dependence on the number of anomalies allowed and relates this hierarchy to $\mathcal{BC}$.

Theorem 21 (Barzdin [16], Case and Smith [27]).

$\mathcal{LIM} \subset \mathcal{LIM}^1 \subset \mathcal{LIM}^2 \subset \cdots \subset \bigcup_{a\in\mathbb{N}} \mathcal{LIM}^a \subset \mathcal{LIM}^* \subset \mathcal{BC}$

For a proof, we refer the reader to Case and Smith [27]. Note that the inclusion $\mathcal{LIM}^* \subset \mathcal{BC}$ appeared already in Barzdin [16]. Thus the option to syntactically change hypotheses entails an *error-correcting power.*

Note that behaviorally correct learning with anomalies has also been studied intensively (cf., e.g., Case and Smith [27], Daley [29], Freivalds [35]). Further results concerning learning with anomalies can be found e.g., in Freivalds *et al.* [42], Gasarch *et al.* [46], Kinber and Zeugmann [62, 61], and Smith and Velauthapillai [90].

Furthermore, using the same ideas as in the proof of Theorem 16 one can easily show the following result.

Theorem 22. *Let* (φ, Φ) *be a complexity measure. There is a function* $s \in \mathcal{R}$ *such that* $\varphi_{s(i)} \in \mathcal{R}$ *and* $\mathcal{BC}_\varphi(\varphi_i) \subseteq \mathcal{BC}_\varphi(\varphi_{s(i)})$ *for all* $i \in \mathbb{N}$.

There is another peculiarity we want to point to. If we sharpen the definition of $\mathcal{BC}$ by adding the requirement that the set $\{S(f^n) \mid n \in \mathbb{N}\}$ of all produced hypotheses is of finite cardinality, then we again get the learning type $\mathcal{LIM}$. This result appeared first in Barzdin and Podnieks [18] and was generalized by Case and Smith [27] (see Theorem 2.9).

A further interesting modification of behaviorally correct learning was introduced by Podnieks [82]. Instead of requiring semantic convergence, he introduced a certain type of uncertainty by demanding correct hypotheses to occur with a certain frequency.

Definition 12 (Podnieks [82]). *Let* $0 < p \leqslant 1$, *let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\varphi \in \mathsf{G\ddot{o}d}$. *The class* $\mathcal{U}$ *is said to be* behaviorally correctly learnable with frequency p *if there is a strategy* $S \in \mathcal{P}$ *such that for each function* $f \in \mathcal{U}$,

(1) *for all* $n \in \mathbb{N}$, $S(f^n)$ *is defined,*

(2) $\liminf\limits_{k\to\infty} \dfrac{|\{n \mid \varphi_{M(f^n)} = f,\ 0 \leqslant n \leqslant k\}|}{k} \geqslant p$

If $\mathcal{U}$ *is behaviorally correctly learnable with frequency* p *by a strategy* S, *we write* $\mathcal{U} \in \mathcal{BC}_{\mathit{freq}}(p)(S)$. $\mathcal{BC}_{\mathit{freq}}(p)$ *is defined analogously to the above.*

Podnieks [82, 83] could prove that $\mathcal{BC}_{\mathit{freq}}\left(\frac{1}{n+1}\right) \subset \mathcal{BC}_{\mathit{freq}}\left(\frac{1}{n+2}\right)$ for all $n \in \mathbb{N}$. Intuitively, this theorem holds, since $\mathcal{BC}$ is not closed under union. For example, taking $\mathcal{U}_0$ and $\mathcal{U}_{sd}$ and trying half the time to learn any function in $\mathcal{U}_0 \cup \mathcal{U}_{sd}$ by simulating any learner for $\mathcal{U}_0$ and for $\mathcal{U}_{sd}$, respectively, and then outputting the hypotheses obtained alternatingly shows that $\mathcal{BC} \subset \mathcal{BC}_{\mathit{freq}}\left(\frac{1}{2}\right)$.

Additionally, he discovered that the $\mathcal{BC}_{\mathit{freq}}$ hierarchy is discrete. More formally, he showed the following. Let p with $1/n \geqslant p > 1/(n+1)$ be given. Then we have $\mathcal{BC}_{\mathit{freq}}(p) = \mathcal{BC}_{\mathit{freq}}\left(\frac{1}{n}\right)$.

Pitt [81] then defined the $\mathcal{LIM}$–analogue to Podnieks' behaviorally correct frequency identification, i.e., $\mathcal{LIM}_{\mathit{freq}}$ and showed an analogous theorem.

Another way to attack the non-closure under finite union has been proposed by Smith [87] who introduced the notion of team learning (or pluralistic inference). For the sake of motivation imagine the task that a robot has to explore a planet. There may be different models for the dynamics of the planet and so the robot is required to learn. While it may be possible to learn the parameters of each single model, due to the non-closure under finite union, it may be impossible to learn the parameters of all these models at once. So, if the number of models is not too large, it may be possible to send a finite number of robots instead of a single one. If one of them learns successfully, the successful robot can perform the exploration.

So, in the basic model of team learning we allow m learning strategies instead of a single one and request, for each $f \in \mathcal{U}$, one of them to be successful. Of course, one can consider teams of $\mathcal{BC}$ learners or teams of $\mathcal{LIM}$ learners. The resulting learning types are denoted by $\mathcal{BC}_{\mathit{team}}(m)$ and $\mathcal{LIM}_{\mathit{team}}(m)$, respectively.

Last but not least, one can also consider probabilistic inference. In this model, it is required that the sequence $S(f^n)_{n\in\mathbb{N}}$ converges with a certain probability p.

This model has been introduced by Freivalds [43] in the setting of finite learning and was then adapted to $\mathcal{BC}$ and $\mathcal{LIM}$ learning. Let us denote the resulting models by $\mathcal{BC}_{prob}(p)$ and $\mathcal{LIM}_{prob}(p)$, respectively.

Pitt [81] obtained the following beautiful unification results. First, he showed that $\mathcal{BC}_{freq}(p) = \mathcal{BC}_{prob}(p)$ and $\mathcal{LIM}_{freq}(p) = \mathcal{LIM}_{prob}(p)$ for every p with $0 < p \leqslant 1$. Thus, probabilistic identification is also discrete. Additionally, he succeeded to prove the following theorem.

Theorem 23 (Pitt [81]).

(1) $\mathcal{BC}_{freq}\left(\frac{1}{n}\right) = \mathcal{BC}_{prob}\left(\frac{1}{n}\right) = \mathcal{BC}_{team}(n)$ for every $n \in \mathbb{N}^+$.

(2) $\mathcal{LIM}_{freq}\left(\frac{1}{n}\right) = \mathcal{LIM}_{prob}\left(\frac{1}{n}\right) = \mathcal{LIM}_{team}(n)$ for every $n \in \mathbb{N}^+$.

Furthermore, Wiehagen, Freivalds and Kinber [101] showed that, with probability close to 1, probabilistic strategies learning in the limit with n mind changes are able to identify function classes which cannot be identified by any deterministic strategy learning in the limit with n mind changes. Additionally, Freivalds, Kinber and Wiehagen [40] studied finite probabilistic learning and probabilistic learning in the limit in nonstandard numberings. In particular, for $I \in \{\mathcal{FIN}, \mathcal{LIM}\}$, they could show that there exist numberings ψ such that, with respect to ψ, no infinite function class can be I–learned deterministically, whereas every class in I is I-learnable with probability $1-\varepsilon$ for every $\varepsilon > 0$.

Since there are some excellent papers treating probabilistic, pluralistic and frequency identification we are not exploring this subject here in more detail. Instead, the interested reader is referred to Ambainis [3], Apsītis *et al.* [8], Daley [29], Pitt [81], Smith [87, 89] and the references therein.

5.2. *Varying the Set of Admissible Strategies*

It should be noted that in Definition 8 no requirement is made concerning the intermediate hypotheses output by strategy S. So, first, we again aim to introduce the consistency requirement already considered in Section 4. However, there are several possibilities to do this. Since a more detailed study of these different possibilities will shed some light on the question of how natural are intuitive postulates, we shall provide a rather complete discussion here. Additionally, in order to make it more interesting we consider the notion of δ–delay, too, which has recently been introduced by Akama and Zeugmann [2].

Definition 13 (Akama and Zeugmann [2]). *Let* $\mathcal{U} \subseteq \mathcal{R}$*, let* $\psi \in \mathcal{P}^2$ *and let* $\delta \in \mathbb{N}$*. The class* $\mathcal{U}$ *is called* consistently learnable in the limit with δ–delay with respect to ψ *if there is a strategy* $S \in \mathcal{P}$ *such that*

(1) $\mathcal{U} \in \mathcal{LIM}_{\psi}(S)$,

(2) $\psi_{S(f^n)}(x) = f(x)$ *for all* $f \in \mathcal{U}$, $n \in \mathbb{N}$ *and all* x *such that* $x + \delta \leqslant n$.

$\mathcal{CONS}_{\psi}^{\delta}(S)$, $\mathcal{CONS}_{\psi}^{\delta}$ *and* $\mathcal{CONS}^{\delta}$ *are defined analogously to the above.*

Note that for $\delta = 0$ we get Barzdin's [10] original definition of $\mathcal{CONS}$. We therefore usually omit the upper index δ if $\delta = 0$. This is also done for all other versions of consistent learning defined below. We use the term δ*–delay*, since a consistent strategy with δ–delay correctly reflects all but at most the last δ data seen so far. If a strategy does not always work consistently with δ–delay we call it δ–delay inconsistent.

Next, we modify $\mathcal{CONS}^{\delta}$ in the same way Jantke and Beick [59] changed $\mathcal{CONS}$, i.e., we add the requirement that the strategy is defined on every input.

Definition 14 (Akama and Zeugmann [2]). *Let* $\mathcal{U} \subseteq \mathcal{R}$*, let* $\psi \in \mathcal{P}^2$ *and let* $\delta \in \mathbb{N}$*. The class* $\mathcal{U}$ *is called* $\mathcal{R}$–consistently learnable in the limit with δ–delay with respect to ψ *if there is a strategy* $S \in \mathcal{R}$ *such that* $\mathcal{U} \in \mathcal{CONS}_{\psi}^{\delta}(S)$*.*

$\mathcal{R}\text{-}\mathcal{CONS}_{\psi}^{\delta}(S)$, $\mathcal{R}\text{-}\mathcal{CONS}_{\psi}^{\delta}$ *and* $\mathcal{R}\text{-}\mathcal{CONS}^{\delta}$ *are defined analogously to the above.*

Note that in Definition 14 consistency with δ–delay is only demanded for inputs that correspond to some function f in the target class. Therefore, in the following definition we incorporate Wiehagen and Liepe's [102] requirement on a strategy to work consistently on all inputs into our scenario of consistency with δ–delay.

Definition 15 (Akama and Zeugmann [2]). *Let* $\mathcal{U} \subseteq \mathcal{R}$*, let* $\psi \in \mathcal{P}^2$ *and let* $\delta \in \mathbb{N}$*. The class* $\mathcal{U}$ *is called* $\mathcal{T}$–consistently learnable in the limit with δ–delay with respect to ψ *if there is a strategy* $S \in \mathcal{R}$ *such that*

(1) $\mathcal{U} \in \mathcal{CONS}_{\psi}^{\delta}(S)$,

(2) $\psi_{S(f^n)}(x) = f(x)$ *for all* $f \in \mathcal{R}$, $n \in \mathbb{N}$ *and all* x *such that* $x + \delta \leqslant n$.

$\mathcal{T}\text{-}\mathcal{CONS}_{\psi}^{\delta}(S)$, $\mathcal{T}\text{-}\mathcal{CONS}_{\psi}^{\delta}$ *and* $\mathcal{T}\text{-}\mathcal{CONS}^{\delta}$ *are defined in the same way as above.*

So, for $\delta = 0$ we again obtain the learning type $\mathcal{T}\text{-}\mathcal{CONS}$ already considered at the end of Section 4.

Next, we introduce *coherent* learning (again with δ-delay). While our consistency with δ-delay demand requires a strategy to correctly reflect all but at most the last δ data seen so far, the coherence requirement only demands to correctly reflect the value $f(n \dot{-} \delta)$ on input f^n.

Definition 16 (Akama and Zeugmann [2]). *Let* $\mathcal{U} \subseteq \mathcal{R}$*, let* $\psi \in \mathcal{P}^2$ *and let* $\delta \in \mathbb{N}$*. The class* $\mathcal{U}$ *is called* coherently learnable in the limit with δ–delay with respect to ψ *if there is a strategy* $S \in \mathcal{P}$ *such that*

(1) $\mathcal{U} \in \mathcal{LIM}_{\psi}(S)$,

(2) $\psi_{S(f^n)}(n \dot{-} \delta) = f(n \dot{-} \delta)$ *for all* $f \in \mathcal{U}$ *and all* $n \in \mathbb{N}$ *such that* $n \geqslant \delta$.

$\mathcal{COH}_{\psi}^{\delta}(S)$, $\mathcal{COH}_{\psi}^{\delta}$ *and* $\mathcal{COH}^{\delta}$ *are defined analogously to the above.*

Now, performing the same modifications to coherent learning with δ–delay as we did in Definitions 14 and 15 to consistent learning with δ–delay results in the learning types $\mathcal{R}\text{-}\mathcal{COH}^{\delta}$ and $\mathcal{T}\text{-}\mathcal{COH}^{\delta}$, respectively. We therefore omit the formal definitions of these learning types here.

Using standard techniques one can show that for all $\delta \in \mathbb{N}$ and all learning types $\mathsf{LT} \in \{\mathcal{CONS}^{\delta}, \mathcal{R}\text{-}\mathcal{CONS}^{\delta}, \mathcal{T}\text{-}\mathcal{CONS}^{\delta}, \mathcal{COH}^{\delta}, \mathcal{R}\text{-}\mathcal{COH}^{\delta}, \mathcal{T}\text{-}\mathcal{COH}^{\delta}\}$ we have $\mathsf{LT}_{\varphi} = \mathsf{LT}$ for every Gödel numbering φ (cf. Lemma 2).

A natural question arising is whether or not the introduction of δ–delay to consistent and coherent learning yields an advantage with respect to the learning power of the defined learning types.

For answering this problem it is advantageous to recall the definition of reliable learning introduced by Blum and Blum [19] and Minicozzi [75]. Intuitively, a learning strategy S is reliable on a set $\mathcal{M}$ provided it converges, when fed the graph of a function f in $\mathcal{M}$, if and only if it learns f. Thus, reliable strategies *signal* their inability to learn by performing infinitely many mind changes.

Definition 17 (Blum and Blum [19], Minicozzi [75]). *Let* $\mathcal{U} \subseteq \mathcal{R}$, *let* $\mathcal{M} \subseteq \mathfrak{P}$ *and let* $\varphi \in \mathsf{Göd}$*; then* $\mathcal{U}$ *is said to be* reliably learnable on $\mathcal{M}$ *if there is a strategy* $S \in \mathcal{R}$ *such that*

(1) $\mathcal{U} \in \mathcal{LIM}_{\varphi}(S)$, *and*

(2) *for all functions* $f \in \mathcal{M}$, *if the sequence* $(S(f^n))_{n \in \mathbb{N}}$ *converges, say to* j, *then* $\varphi_j = f$.

$\mathcal{M}\text{-}\mathcal{REL}$ *denotes the family of all function classes that are reliably learnable on* $\mathcal{M}$.

In particular, we shall consider the cases where $\mathcal{M} = \mathfrak{T}$ and $\mathcal{M} = \mathcal{R}$, i.e., reliable learnability on the set of all total functions and all recursive functions, respectively. For the sake of completeness, we also mention here that the family of all function classes reliably identifiable on the set of all partial functions equals the set of all function classes reliably learnable on the set of all partial recursive functions. Furthermore, reliable learning on the set of all partial functions allows the following characterization in terms of consistency.

Theorem 24 (Blum and Blum [19]). $\mathfrak{P}\text{-}\mathcal{REL} = \mathcal{P}\text{-}\mathcal{REL} = \mathcal{T}\text{-}\mathcal{CONS}^{arb}$.

Moreover, reliable learning possesses some very nice closure properties as shown by Minicozzi [75] (cf. Theorems 3 and 4 in [75]). For the sake of completeness, we recall these results here but refer the reader to [75] for a proof.

Theorem 25 (Minicozzi [75]). *Let* $\mathcal{M} \subseteq \mathfrak{P}$*; then we have:*

(1) $\mathcal{M}\text{-}\mathcal{REL}$ *is closed under recursively enumerable union.*

(2) *For every class* $\mathcal{U} \subseteq \mathcal{R}$, *if* $\mathcal{U} \in \mathcal{M}\text{-}\mathcal{REL}$ *then also the class of all finite variants of the functions in* $\mathcal{U}$ *is reliable learnable on* $\mathcal{M}$, *i.e.*, $[[\mathcal{U}]] \in \mathcal{M}\text{-}\mathcal{REL}$.

The following theorem provides a first insight into the learning capabilities of reliable learning in dependence on the set $\mathcal{M}$. The first rigorous proof of $\mathcal{T}\text{-}\mathcal{REL} \subset \mathcal{R}\text{-}\mathcal{REL}$ appeared in Grabowski [49]. A conceptually much easier proof has been provided by Stephan and Zeugmann [94]. Therefore, we skip this proof below.

Theorem 26. $\mathcal{P}\text{-}\mathcal{REL} \subset \mathcal{T}\text{-}\mathcal{REL} \subset \mathcal{R}\text{-}\mathcal{REL} \subset \mathcal{LIM}$.

Proof. $\mathcal{P}\text{-}\mathcal{REL} \subset \mathcal{T}\text{-}\mathcal{REL}$ is a direct consequence of Theorems 15 and 24.

$\mathcal{R}\text{-}\mathcal{REL} \subseteq \mathcal{LIM}$ is obvious. For showing that $\mathcal{LIM} \setminus \mathcal{R}\text{-}\mathcal{REL} \neq \emptyset$, we use the class $\mathcal{U}_{sd}$ which is clearly in $\mathcal{LIM}$. Suppose $\mathcal{U}_{sd} \in \mathcal{R}\text{-}\mathcal{REL}$. Then applying Theorem 25 directly yields $[[\mathcal{U}_{sd}]] \in \mathcal{R}\text{-}\mathcal{REL}$, too. But $[[\mathcal{U}_{sd}]] = \mathcal{R}$ (cf. Claim 2 in the proof of Lemma 1). Since $\mathcal{R}\text{-}\mathcal{REL} \subseteq \mathcal{LIM}$, we get $\mathcal{R} \in \mathcal{LIM}$, a contradiction to Corollary 20. ∎

Note that one can extend the notion of reliable learning to behaviorally correct reliable inference, too. Additionally, starting from the notion of reliability one can define for $\mathcal{BC}$– and $\mathcal{LIM}$–type identification the notion of one-sided error probabilistic learning as well as of reliable frequency identification (see Kinber and Zeugmann [61]). The flavor of the obtained results is similar to Podnieks' [82, 83] and Pitt's [81]. On the other hand, one can also look at team learning as a way of introducing a bounded nondeterminism to learning. But even introducing an unbounded nondeterminism to reliable learning does not enlarge the learning capabilities of reliable $\mathcal{LIM}$ inference (see Pitt [81], Theorem 4.14). So, though we have Theorem 23, there are subtle differences between probabilistic and frequency identification on the one hand and pluralistic learning on the other.

5.3. *Varying the Information Supply*

Next, we consider two variations of the information fed to the learner. Looking at all the learning models defined so far we see that a strategy has always access to all examples presented so far. In the following definition, we consider the variant where the strategy is only allowed to use its last guess and the new datum coming in.

Definition 18 (Wiehagen [97]). *Let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\psi \in \mathcal{P}^2$. *The class* $\mathcal{U}$ *is said to be* iteratively learnable with respect to ψ *if there is a strategy* $S \in \mathcal{P}$ *such that for each function* $f \in \mathcal{U}$,

(1) *for every* $n \in \mathbb{N}$, $S_n(f)$ *is defined, where*

$$S_0(f) = S(0, f(0)), \text{ and}$$

$$S_{n+1}(f) = S(S_n(f), n+1, f(n+1)).$$

(2) *There is a* $j \in \mathbb{N}$ *such that* $\psi_j = f$ *and the sequence* $(S(f^n))_{n \in \mathbb{N}}$ *converges to* j.

If the class $\mathcal{U}$ *is iteratively learnable with respect to* ψ *by a strategy* S, *we write* $\mathcal{U} \in \mathcal{IT}_\psi(S)$. *Furthermore,* $\mathcal{IT}_\psi$ *and* $\mathcal{IT}$ *are defined analogously to the above.*

Of course, an iterative strategy can try to memorize the pairs $(n, f(n))$ in its current hypothesis. Then the strategy would have access to the whole initial segment f^n presented so far. On the other hand, the strategy has to converge. Therefore, an iterative strategy can only memorize *finitely* many pairs $(n, f(n))$, i.e., a finite subfunction, in its hypothesis. Consequently, it is only natural to ask whether or not this restriction does decrease the resulting learning power. The affirmative answer is provided by the following theorem.

Theorem 27 (Wiehagen [97]). $\mathcal{IT} \subset \mathcal{LIM}$

Proof. Clearly, we have $\mathcal{IT} \subseteq \mathcal{LIM}$. It remains to show that $\mathcal{LIM} \setminus \mathcal{IT} \neq \emptyset$. The separating class $\mathcal{U}$ is defined as follows. We modify the class of self-describing functions a bit by requiring all function values to be strictly positive, i.e., we set $\mathcal{U}_{sdp} = \{f \mid f \in \mathcal{R},\ \varphi_{f(0)} = f,\ \forall x[f(x) > 0]\}$ and $\mathcal{U} = \mathcal{U}_0 \cup \mathcal{U}_{sdp}$.

Claim 1. $\mathcal{U} \in \mathcal{LIM}$

Intuitively, the desired strategy S outputs $f(0)$ as long as all function values seen so far are greater than 0. If S sees 0 as a function value for the first time, it switches its learning mode. From this point onwards S uses the identification by enumeration strategy to learn the target function. We omit the details.

Claim 2. $\mathcal{U} \notin \mathcal{IT}$

It suffices to show that for every S with $\mathcal{U}_0 \in \mathcal{IT}_\varphi(S)$ there is a function $f \in \mathcal{U}_{sdp}$ such that $f \notin \mathcal{IT}_\varphi(S)$. Let $s \in \mathcal{R}$ be chosen such that for all $j \in \mathbb{N}$

$$\begin{aligned}
\varphi_{s(j)}(0) &= j, \text{ and for all } n \in \mathbb{N}: \\
\varphi_{s(j)}(n+1) &= \begin{cases} 1, & \text{if } S(S_n(\varphi_{s(j)}), n+1, 1) \neq S_n(\varphi_{s(j)}) \\ 2, & \text{if } S(S_n(\varphi_{s(j)}), n+1, 1) = S_n(\varphi_{s(j)}) \text{ and} \\ & \quad S(S_n(\varphi_{s(j)}), n+1, 2) \neq S_n(\varphi_{s(j)}) \end{cases}
\end{aligned}$$

Note that one of these cases must happen. For seeing this, suppose the converse. Let m be the least n such that

$$S(S_{n-1}(\varphi_{s(j)}), n, 1) = S(S_{n-1}(\varphi_{s(j)}), n, 2) = S_{n-1}(\varphi_{s(j)}) \ .$$

Now consider the functions g and g' defined as

$$g(x) = \begin{cases} \varphi_{s(j)}(x), & \text{if } x < m \\ 1 \qquad\quad , & \text{if } x = m \\ 0 \qquad\quad , & \text{if } x > m \ , \end{cases}$$

and $g'(x) = g(x)$ for all $x \neq m$ and $g'(m) = 2$. Since $g,\ g' \in \mathcal{U}_0$ the strategy S must iteratively learn both g and g'. But by the choice of m we can directly conclude that the sequences $(S_n(g))_{n \in \mathbb{N}}$ and $(S_n(g'))_{n \in \mathbb{N}}$ converge to the same number, a contradiction.

Consequently, $\varphi_{s(j)} \in \mathcal{R}$ for every j. By the fixed point theorem (cf., e.g., [88]) there is an $i \in \mathbb{N}$ such that $\varphi_i = \varphi_{s(i)}$. By construction, $\varphi_i \in \mathcal{U}_{sdp}$ and S changes its hypothesis in every learning step when successively fed φ_i. Thus, for $f = \varphi_i$ we have $f \notin \mathcal{IT}_\varphi(S)$. ∎

It should be mentioned that Wiehagen [97] proved a slightly stronger result than our Theorem 27, since he showed the class $\mathcal{U}$ in the proof above to be even learnable by a feed-back strategy. A *feed-back* strategy, when successively fed a function f works like an iterative strategy but can additionally make a query by computing an argument x and asking for $f(x)$. While feed-back learning is stronger than iterative learning, it is still weaker than learning in the limit. It should be noted that a suitably modified version of feed-back learning has recently attracted attention in the setting of language learning from positive data (see [24, 68]).

Finally, iterative learning is also quite sensitive to the order in which examples are presented. Jantke and Beick [59] considered $\mathcal{IT}^{arb}$ and showed the following result.

Theorem 28 (Jantke and Beick [59]). $\mathcal{R}\text{-}\mathcal{TOTAL} \# \mathcal{IT}^{arb}$

In the next definition, we consider a variant of how to enrich the information presented to a learner. This type of inference has been introduced by Wiehagen [99] and has been intensively studied in Freivald and Wiehagen [32]. It has been further investigated by Freivalds, Botuscharov, and Wiehagen [38] and, in the context of language identification, by Jain and Sharma [56]. We refer to it as learning with additional information.

Definition 19 (Wiehagen [99]). *Let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\varphi \in \mathsf{G\ddot{o}d}$. $\mathcal{U} \in \mathcal{LIM}^+$ *if there is a strategy* $S \in \mathcal{P}^2$ *such that for every* $f \in \mathcal{U}$ *and for every bound* $s \geqslant \min_\varphi f$ *the following conditions are satisfied.*

(1) $S(s, f^n)$ *is defined for all* $n \in \mathbb{N}$, *and*

(2) *the sequence* $(S(s, f^n))_{n\in\mathbb{N}}$ *converges to a number* j *such that* $\varphi_j = f$.

Whenever appropriate, we shall also consider LT^+ for any of the learning types defined in this paper.

Learning with additional information shows that consistent learning is full of surprises. Note that Assertion (1) in the following theorem has been shown by Freivald and Wiehagen [32], while Assertion (2) goes back to Wiehagen [99].

Theorem 29 (Freivald and Wiehagen [32], Wiehagen [99]).

(1) $\mathcal{T}\text{-}\mathcal{CONS}^+ = \mathcal{T}\text{-}\mathcal{CONS}$, *and*

(2) $\mathcal{R} \in \mathcal{CONS}^+$.

Proof. Since we obviously have $\mathcal{T}\text{-}\mathcal{CONS} \subseteq \mathcal{T}\text{-}\mathcal{CONS}^+$, it suffices to show that $\mathcal{T}\text{-}\mathcal{CONS}^+ \subseteq \mathcal{T}\text{-}\mathcal{CONS}$. Let $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}^+(S)$, where $S \in \mathcal{R}^2$. Then for every $f \in \mathcal{U}$ we can construct in the limit a number s such that the sequence $(S(s, f^n))_{n\in\mathbb{N}}$ converges to a number j. Since S is $\mathcal{T}$-consistent, we can conclude that $\varphi_j = f$. Note that, in general, we do not have $s \geqslant \min_\varphi f$. The formal proof is done as follows. We have to define a strategy $S' \in \mathcal{R}$ such that $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}(S')$. Let $\alpha \in \mathbb{N}^*$ be any tuple of length 1. The desired strategy S' is defined as follows.

We set $i_0 = 0$ and $S'(\alpha) = S(i_0, \alpha)$.

Now assume $n \in \mathbb{N}$ such that i_n and $S'(\alpha)$ for all tuples of length $n+1$ are already defined. Let $y \in \mathbb{N}$; we set

$i_{n+1} = i_n$ and $S'(\alpha y) = S(i_n, \alpha y)$ provided $S(i_n, \alpha y) = S(i_n, \alpha)$. Otherwise, we set $i_{n+1} = i_n + 1$ and $S'(\alpha y) = S(i_{n+1}, \alpha y)$.

By construction, we directly obtain $S' \in \mathcal{R}$ because of $S \in \mathcal{R}^2$. Furthermore, S is $\mathcal{T}$-consistent, so is S'. Additionally, since for every $f \in \mathcal{U}$ there is an $s \in \mathbb{N}$ such that the sequence $(S(s, f^n))_{n\in\mathbb{N}}$ converges (every $s \geqslant \min_\varphi f$ has this property) the sequence $(S'(f^n))_{n\in\mathbb{N}}$ must converge, too. So, let j be the number the sequence $(S'(f^n))_{n\in\mathbb{N}}$ converges to. Finally, by the $\mathcal{T}$-consistency of S' we can conclude that $\varphi_j = f$. This proves Assertion (1).

For showing the remaining Part (2), we use the amalgamation technique (cf. Wiehagen [99], Case and Smith [27]). Let amal be a recursive function mapping any finite set I of φ-programs to a φ-program such that for any $x \in \mathbb{N}$, $\varphi_{\mathrm{amal}(I)}(x)$ is defined by running $\varphi_i(x)$ for every $i \in I$ in parallel and taking the first value obtained, if any. The desired strategy $S \in \mathcal{P}^2$ is mainly defined by using the function amal defined above. Let $f \in \mathcal{R}$ and let $s \in \mathbb{N}$; we set $I_{f,-1} = \{0, \ldots, s\}$. For $n \geqslant 0$ we proceed inductively. Assume $I_{f,n-1}$ to be already defined. We set

$$t \;=\; \text{“the minimal number such that for all } 0 \leqslant x \leqslant n \text{ there is an } i \in I_{f,n-1} \text{ with } \Phi_i(x) \leqslant t \text{ and } \varphi_i(x) = f(x)\text{ .”}$$

Furthermore, we define

$$I^-_{f,n} = \{i \mid i \in I_{f,n-1},\ \exists x \leqslant n[\Phi_i(x) \leqslant t,\ \varphi_i(x) \neq f(x)]\}\ .$$

Moreover, we set $I_{f,n} = I_{f,n-1} \setminus I^-_{f,n}$. Now we define the desired strategy $S \in \mathcal{P}^2$ as follows. For all $n \in \mathbb{N}$ and all $s \in \mathbb{N}$ let

$S(s, f^n) =$ “Compute $I_{f,n}$. If the computation of $I_{f,n}$ stops then let
$S(s, f^n) = \mathrm{amal}(I_{f,n})$.
Otherwise, $S(s, f^n) = \uparrow$.”

It remains to show that $\mathcal{R} \in \mathcal{CONS}^+(S)$. Let $s \in \mathbb{N}$ be any number such that $s \geqslant \min_\varphi f$. Then, by construction, the computation of $I_{f,n}$ stops for all $n \in \mathbb{N}$ and we have $I_{f,n} \subseteq I_{f,n-1}$ for all $n \in \mathbb{N}$. Furthermore, by construction S is consistent, too. Since $\min_\varphi f \in I_{f,n}$ for all $n \in \mathbb{N}$, we also have $I_{f,n} \neq \emptyset$ for all $n \in \mathbb{N}$. Consequently, the sequence $(I_{f,n})_{n\in\mathbb{N}}$ of sets converges to a finite and non-empty set I containing at least one φ-program for f. Thus, the sequence $(S(s, f^n))_{n\in\mathbb{N}}$ converges to amal(I) and since S is consistent we can conclude $\varphi_{\mathrm{amal}(I)} = f$. This proves Assertion (2). ∎

Another interesting effect is observed when studying $\mathcal{FIN}^+$. In contrast to Theorem 17, the learning type $\mathcal{FIN}^+$ comprises classes containing an accumulation point, e.g., $\mathcal{U} = \{0^i10^\infty \mid i \leqslant \min_\varphi 0^i10^\infty\} \cup \{0^\infty\}$. On the other hand, it is easy to show that $\{0^i10^\infty \mid i \in \mathbb{N}\} \cup \{0^\infty\} \notin \mathcal{FIN}^+$. Thus, we directly get:

Theorem 30. $\mathcal{FIN} \subset \mathcal{FIN}^+ \subset \wp(\mathcal{R})$.

For further information concerning inductive inference with additional information, we refer the interested reader to Jain *et al.* [55].

Before exploring further relations between consistent and reliable learning we take a closer look at coherent and consistent identification.

5.4. *Coherence and Consistency of Learning Strategies*

In this subsection we study the question whether or not the relaxation to learn coherently with δ–delay instead of demanding consistency with δ–delay does enhance the learning power of the corresponding learning types introduced above. The negative answer is provided by the following theorem.

Theorem 31 (Akama and Zeugmann [2]). *Let* $\delta \in \mathbb{N}$ *be arbitrarily fixed. Then we have*

(1) $\mathcal{CONS}^\delta = \mathcal{COH}^\delta$,

(2) $\mathcal{R}\text{-}\mathcal{CONS}^\delta = \mathcal{R}\text{-}\mathcal{COH}^\delta$,

(3) $\mathcal{T}\text{-}\mathcal{CONS}^\delta = \mathcal{T}\text{-}\mathcal{COH}^\delta$.

Proof. By definition, we obviously have $\mathcal{CONS}^\delta \subseteq \mathcal{COH}^\delta$, $\mathcal{R}\text{-}\mathcal{CONS}^\delta \subseteq \mathcal{R}\text{-}\mathcal{COH}^\delta$ and $\mathcal{T}\text{-}\mathcal{CONS}^\delta \subseteq \mathcal{T}\text{-}\mathcal{COH}^\delta$.

For showing the opposite directions we can essentially use the same idea for all three cases. Let $\delta \in \mathbb{N}$, $\varphi \in \mathsf{G\ddot{o}d}$, $\mathcal{U} \subseteq \mathcal{R}$ and any strategy $\hat{S}$ be arbitrarily fixed such that $\mathcal{U} \in \mathsf{LT}_\varphi(\hat{S})$, where $\mathsf{LT} \in \{\mathcal{COH}^\delta,\ \mathcal{R}\text{-}\mathcal{COH}^\delta,\ \mathcal{T}\text{-}\mathcal{COH}^\delta\}$. Next, we define a strategy S as follows. Let $f \in \mathcal{R}$ and let $n \in \mathbb{N}$. On input f^n do the following.

1. Compute $\hat{S}(f^0), \ldots, \hat{S}(f^n)$ and determine the largest number n^* such that $\hat{S}(f^{n^*-1}) \neq \hat{S}(f^{n^*})$.
2. Output the canonical φ–program i computing the following function g:

   $g(x) = f(x)$ for all $x \leqslant n^*$, and

   $g(x) = \varphi_{\hat{S}(f^{n^*})}(x)$ for all $x > n^*$.

First, we show that S learns $\mathcal{U}$ consistently with δ–delay.

By construction, we have $\varphi_{S(f^n)}(x) = f(x)$ for all $x \leqslant n^*$, and thus S is consistent on $f(0), \ldots, f(n^*)$. If $n - n^* \leqslant \delta$, we are already done. Finally, if $n - n^* > \delta$, then we exploit the fact that $\hat{S}$ works coherently with δ–delay and that $\hat{S}(f^{n^*+k}) = \hat{S}(f^{n^*})$ for all $k = 1, \ldots, n - n^*$. Thus, for all $k \in \{1, \ldots, n - n^* - \delta\}$ we get

$$\varphi_{S(f^n)}(n^* + k) \ = \ \varphi_{\hat{S}(f^{n^*})}(n^* + k) = \varphi_{\hat{S}(f^{n^*+\delta+k})}(n^* + k) = f(n^* + k)\ . \qquad (2)$$

Since in this case $\hat{S}(f^n)$ is defined for all $f \in \mathcal{U}$ and all $n \in \mathbb{N}$, we can directly conclude that $S(f^n)$ is defined for all $f \in \mathcal{U}$ and all $n \in \mathbb{N}$, too. This proves Assertion (1).

If $\hat{S} \in \mathcal{R}$, then so is S and thus Assertion (2) follows.

Finally, if $\hat{S} \in \mathcal{R}$ and $\hat{S}$ works $\mathcal{T}$–coherently, then we directly get $S \in \mathcal{R}$ and S is $\mathcal{T}$–consistent, since now (2) is true for all $f \in \mathcal{R}$. This completes the proof. ▮

6. Characterizations in Terms of Complexity

In this section we characterize $\mathcal{T}\text{-}\mathcal{CONS}^\delta$, $\mathcal{CONS}^\delta$, $\mathfrak{T}\text{-}\mathcal{REL}$, $\mathcal{R}\text{-}\mathcal{REL}$, and $\mathcal{LIM}$ in terms of complexity. The importance of such characterizations has already been explained in Subsection 3.1. However, in order to achieve the aforementioned characterizations, several modifications are necessary. In particular, so far we used functions to compute the relevant complexity bounds in the definitions of the complexity classes $\mathcal{R}_t$, where $t \in \mathcal{R}$ and in $\mathcal{R}_h$, where $h \in \mathcal{R}^2$. Now we need stronger tools, i.e., computable operators which are introduced next.

First, we recall the definitions of recursive and general recursive operator. Let $(F_x)_{x\in\mathbb{N}}$ be the canonical enumeration of all finite functions.

Definition 20 (Rogers [86]). *A mapping* $\mathfrak{O}\colon \mathfrak{P} \mapsto \mathfrak{P}$ *from partial functions to partial functions is called a* partial recursive operator *if there is a recursively enumerable set* $W \subseteq \mathbb{N}^3$ *such that for any* $y, z \in \mathbb{N}$ *it holds that* $\mathfrak{O}(f)(y) = z$ *iff there is an* $x \in \mathbb{N}$ *such that* $(x, y, z) \in W$ *and* f *extends the finite function* F_x*.*

Furthermore, a partial recursive operator $\mathfrak{O}$ *is called a* general recursive operator *iff* $\mathfrak{T} \subseteq \mathtt{dom}(\mathfrak{O})$*, and* $f \in \mathfrak{T}$ *implies* $\mathfrak{O}(f) \in \mathfrak{T}$*.*

A mapping $\mathfrak{O}\colon \mathcal{P} \mapsto \mathcal{P}$ *is called an* effective operator *iff there is a function* $g \in \mathcal{R}$ *such that* $\mathfrak{O}(\varphi_i) = \varphi_{g(i)}$ *for all* $i \in \mathbb{N}$*. An effective operator* $\mathfrak{O}$ *is said to be* total effective *provided that* $\mathcal{R} \subseteq \mathtt{dom}(\mathfrak{O})$*, and* $\varphi_i \in \mathcal{R}$ *implies* $\mathfrak{O}(\varphi_i) \in \mathcal{R}$*.*

For more information about general recursive operators and effective operators the reader is referred to [52, 77, 107]. If $\mathfrak{O}$ is an operator which maps functions to functions, we write $\mathfrak{O}(f, x)$ to denote the value of the function $\mathfrak{O}(f)$ at the argument x. Any computable operator can be realized by a 3-tape Turing machine T which works as follows: If for an arbitrary function $f \in \mathtt{dom}(\mathfrak{O})$, all pairs $(x, f(x))$, $x \in \mathtt{dom}(f)$ are written down on the input tape of T (repetitions are allowed), then T will write exactly all pairs $(x, \mathfrak{O}(f, x))$ on the output tape of T (under unlimited working time).

Let $\mathfrak{O}$ be a general recursive or total effective operator. Then, for $f \in \mathtt{dom}(\mathfrak{O})$, $m \in \mathbb{N}$, we set: $\Delta\mathfrak{O}(f, m) =$"the least n such that, for all $x \leqslant n$, $f(x)$ is defined and, for the computation of $\mathfrak{O}(f, m)$, the Turing machine T only uses the pairs $(x, f(x))$ with $x \leqslant n$; if such an n does not exist, we set $\Delta\mathfrak{O}(f, m) = \infty$."

For $u \in \mathcal{R}$ we define Ω_u to be the set of all partial recursive operators $\mathfrak{O}$ satisfying $\Delta\mathfrak{O}(f, m) \leqslant u(m)$ for all $f \in \mathtt{dom}(\mathfrak{O})$. For the sake of notation, below we shall use $\mathtt{id} + \delta$, $\delta \in \mathbb{N}$, to denote the function $u(x) = x + \delta$ for all $x \in \mathbb{N}$.

Now we are ready to provide the first group of characterizations.

6.1. *Characterizing* $\mathcal{T}\text{-}\mathcal{CONS}^\delta$ *and* $\mathcal{CONS}^\delta$

Note that in the following we use mainly ideas and techniques from Blum and Blum [19] and Wiehagen [99].

Furthermore, in the following we always assume that learning is done with respect to any fixed $\varphi \in \mathsf{Göd}$.

As in Blum and Blum [19] we define operator complexity classes as follows. Let $\mathfrak{O}$ be any computable operator; then we set

$$\mathcal{R}_{\mathfrak{O}} = \{f \mid \exists i[\varphi_i = f \wedge \forall^\infty x[\Phi_i(x) \leqslant \mathfrak{O}(f,x)]]\} \cap \mathcal{R} .$$

First, we characterize $\mathcal{T}\text{-}\mathcal{CONS}^\delta$.

Theorem 32. *Let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\delta \in \mathbb{N}$*; then we have:* $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}^\delta$ *if and only if there exists a general recursive operator* $\mathfrak{O} \in \Omega_{id+\delta}$ *such that* $\mathfrak{O}(\mathcal{R}) \subseteq \mathcal{R}$ *and* $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$.

Proof. Necessity. Let $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}^\delta(S)$, $S \in \mathcal{R}$. Then for all $f \in \mathcal{R}$ and all $n \in \mathbb{N}$ we define $\mathfrak{O}(f,n) = \Phi_{S(f^{n+\delta})}(n)$.

Since $\varphi_{S(f^{n+\delta})}(n)$ is defined for all $f \in \mathcal{R}$ and all $n \in \mathbb{N}$ by Condition (2) of Definition 15, we directly get from Condition (1) of the definition of a complexity measure that $\Phi_{S(f^{n+\delta})}(n)$ is defined for all $f \in \mathcal{R}$ and all $n \in \mathbb{N}$, too. Moreover, for every $t \in \mathfrak{T}$ and $n \in \mathbb{N}$ there is an $f \in \mathcal{R}$ such that $t^n = f^n$. Hence, we have $\mathfrak{O}(\mathfrak{T}) \subseteq \mathcal{R} \subseteq \mathfrak{T}$. Moreover, in order to compute $\mathfrak{O}(f,n)$ the operator $\mathfrak{O}$ reads only the values $f(0), \dots, f(n+\delta)$. Thus, we have $\mathfrak{O} \in \Omega_{id+\delta}$.

Now let $f \in \mathcal{U}$. Then the sequence $(S(f^n))_{n\in\mathbb{N}}$ converges to a correct φ–program i for f. Consequently, $\mathfrak{O}(f,n) = \Phi_i(n)$ for almost all $n \in \mathbb{N}$. Therefore, we conclude $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$.

Sufficiency. Let $\mathfrak{O} \in \Omega_{id+\delta}$ such that $\mathfrak{O}(\mathcal{R}) \subseteq \mathcal{R}$ and $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$. We have to define a strategy $S \in \mathcal{R}$ such that $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}^\delta(S)$. By the definition of $\mathcal{R}_{\mathfrak{O}}$ we know that for every $f \in \mathcal{U}$ there exist i and k such that $\varphi_i = f$ and $\Phi_i(x) \leqslant \max\{k,\ \mathfrak{O}(f,x)\}$ for all x. Thus, the desired strategy S searches for the first current candidate for such a pair (i,k) in the canonical enumeration c_2 of $\mathbb{N} \times \mathbb{N}$ and converges to i provided an appropriate pair has indeed been found. Until this pair (i,k) is found, the strategy S outputs auxiliary consistent hypotheses. For doing this, we choose $g \in \mathcal{R}$ such that $\varphi_{g(\langle\alpha\rangle)}(x) = y_x$ for every tuple $\alpha \in \mathbb{N}^*$, $\alpha = (y_0, \dots, y_n)$ and all $x \leqslant n$.

$S(f^n) =$ "Compute $\mathfrak{O}(f,x)$ for all $x \leqslant n \dot{-} \delta$. Search for the least $z \leqslant n$ such that for $c_2(z) = (i,k)$ the conditions

(A) $\Phi_i(x) \leqslant \max\{k,\ \mathfrak{O}(f,x)\}$ for all $x \leqslant n \dot{-} \delta$, and

(B) $\varphi_i(x) = f(x)$ for all $x \leqslant n \dot{-} \delta$

are fulfilled. If such a z is found, set $S(f^n) = i$.

Otherwise, set $S(f^n) = g(f^n)$."

Since $\mathfrak{O} \in \Omega_{id+\delta}$, the strategy can compute $\mathfrak{O}(f, x)$ for all $x \leqslant n \dot{-} \delta$ and since $c_2 \in \mathcal{R}$ it also can perform the desired search effectively. By Condition (2) of the definition of a complexity measure, the test in (A) can be performed effectively, too. If this test has succeeded, then Test (B) can also be effectively executed by Condition (1) of the definition of a complexity measure. Thus, we get $S \in \mathcal{R}$. Finally, by construction S is always consistent with δ-delay, and if $f \in \mathcal{U}$, then the sequence $(S(f^n))_{n\in\mathbb{N}}$ converges to a correct φ–program for f. ∎

Theorem 33. *Let $\mathcal{U} \subseteq \mathcal{R}$ and let $\delta \in \mathbb{N}$; then we have: $\mathcal{U} \in \mathcal{CONS}^\delta$ if and only if there exists a partial recursive operator $\mathfrak{O} \in \Omega_{id+\delta}$ such that $\mathfrak{O}(\mathcal{U}) \subseteq \mathcal{R}$ and $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$.*

Proof. The necessity is proved *mutatis mutandis* as in the proof of Theorem 32 with the only modification that $\mathfrak{O}(f, x)$ is now defined for all $f \in \mathcal{U}$ instead of for all $f \in \mathcal{R}$. This directly yields $\mathfrak{O} \in \Omega_{id+\delta}$, $\mathfrak{O}(\mathcal{U}) \subseteq \mathcal{R}$ and $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$.

The only modification for the sufficiency part is to leave $S(f^n)$ undefined if $\mathfrak{O}(f, x)$ is not defined for $f \notin \mathcal{U}$. We omit the details. ∎

We continue this section by using Theorem 32 to show that $\mathcal{T}\text{-}\mathcal{CONS}^\delta$ is closed under enumerable unions.

Theorem 34. *Let $\delta \in \mathbb{N}$ and let $(S_i)_{i\in\mathbb{N}}$ be a recursive enumeration of strategies working $\mathcal{T}$-consistently with δ-delay. Then there exists a strategy $S \in \mathcal{R}$ such that $\bigcup_{i\in\mathbb{N}} \mathcal{T}\text{-}\mathcal{CONS}^\delta(S_i) \subseteq \mathcal{T}\text{-}\mathcal{CONS}^\delta(S)$.*

Proof. The proof of the necessity of Theorem 32 shows that the construction of the operator $\mathfrak{O}$ is effective provided a program for the strategy is given. Thus, we effectively obtain a recursive enumeration $(\mathfrak{O}_i)_{i\in\mathbb{N}}$ of operators $\mathfrak{O}_i \in \Omega_{id+\delta}$ such that $\mathfrak{O}_i(\mathcal{R}) \subseteq \mathcal{R}$ and $\mathcal{T}\text{-}\mathcal{CONS}^\delta(S_i) \subseteq \mathcal{R}_{\mathfrak{O}_i}$.

Now we define an operator $\mathfrak{O}$ as follows. Let $f \in \mathcal{R}$ and $x \in \mathbb{N}$. We set $\mathfrak{O}(f, x) = \max\{\mathfrak{O}_i(f, x) \mid i \leqslant x\}$.

Thus, we directly get $\mathfrak{O} \in \Omega_{id+\delta}$, $\mathfrak{O}(\mathcal{R}) \subseteq \mathcal{R}$ and $\bigcup_{i\in\mathbb{N}} \mathcal{T}\text{-}\mathcal{CONS}^\delta(S_i) \subseteq \mathcal{R}_{\mathfrak{O}}$. By Theorem 32 we can conclude $\bigcup_{i\in\mathbb{N}} \mathcal{T}\text{-}\mathcal{CONS}^\delta(S_i) \subseteq \mathcal{T}\text{-}\mathcal{CONS}^\delta(S)$. ∎

On the other hand, $\mathcal{CONS}^\delta$ and $\mathcal{R}\text{-}\mathcal{CONS}$ are not even closed under finite union. This is a direct consequence of Theorem 19. It is easy to verify that $\mathcal{U}_{sd}$, $\mathcal{U}_0 \in \mathcal{R}\text{-}\mathcal{CONS}^\delta$ and thus $\mathcal{U}_{sd}$, $\mathcal{U}_0 \in \mathcal{CONS}^\delta$ for every $\delta \in \mathbb{N}$. But $\mathcal{U}_{sd} \cup \mathcal{U}_0 \notin \mathcal{BC}$.

6.2. *Characterizing $\mathcal{T}$-$\mathcal{REL}$, $\mathcal{R}$-$\mathcal{REL}$ and $\mathcal{LIM}$*

We continue with the characterizations of $\mathfrak{T}$-$\mathcal{REL}$, $\mathcal{R}$-$\mathcal{REL}$ and $\mathcal{LIM}$ in terms of complexity. As the following theorem shows, these characterizations express the difference of these learning models by different sets of admissible operators, i.e., general recursive, total effective and effective operators, respectively. Assertion (1) has been shown by Grabowski [49], Assertion (2) by Blum and Blum [19] and Assertion (3) is a variation of a corresponding characterization obtained by Wiehagen [99].

In the proofs below, it is technically convenient to use limiting recursive functionals instead of partial recursive functions as strategies. For a formal machine independent definition of a limiting recursive functional see Rogers [86]. Intuitively, a limiting recursive functional is a mapping which maps functions to numbers in a computable way. Using 3-tape Turing machines with input, work and output tape and a read-only head for the input tape, a read-write head for the work tape and a write-only head for the output tape, a limiting recursive functional can be defined as follows.

A partial mapping $\mathsf{S}\colon \mathfrak{P} \mapsto \mathbb{N}$ is called limiting recursive functional if there is a 3-tape Turing machine T (as described above) working as follows:

If an arbitrary function $\mathsf{f} \in \mathfrak{P}$ is written down on the input tape of T (in an arbitrary enumeration of input-output examples where repetitions are allowed), then, if $\mathsf{S}(\mathsf{f})$ is defined, T writes a finite nonempty sequence of natural numbers on the output tape such that the last number is equal to $\mathsf{S}(\mathsf{f})$; (T does not need to stop after doing so), or T writes an infinite sequence of natural numbers which converges on its output tape such that its limit is equal to $\mathsf{S}(\mathsf{f})$. It is allowed that the sequence written on the output tape depends on the enumeration in which the function f is written on the input tape, but it is prohibited that its limit depends on it.

If $\mathsf{S}(\mathsf{f})$ is not defined, then two cases are possible. First, S does not uniformly converge on some enumeration in which the function f is written on the input tape. Second, S never converges – independent of the enumeration in which the function f is written on the input tape. These cases are not equivalent (cf. Freivald [33]). Therefore, we require that for all $\mathsf{f} \in \mathfrak{P}$ we have: $\mathsf{f} \notin \mathtt{dom}(\mathsf{S})$ iff S on f never converges.

Theorem 35. *Let $\mathcal{U} \subseteq \mathcal{R}$, then we have:*

(1) $\mathcal{U} \in \mathfrak{T}$-$\mathcal{REL}$ *if and only if there exists a general recursive operator $\mathfrak{O}$ such that $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$.*

(2) $\mathcal{U} \in \mathcal{R}$-$\mathcal{REL}$ *if and only if there exists a total effective operator $\mathfrak{O}$ such that $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$.*

(3) $\mathcal{U} \in \mathcal{LIM}$ *if and only if there exists an effective operator $\mathfrak{O}$ such that $\mathfrak{O}(\mathcal{U}) \subseteq \mathcal{R}$ and $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$.*

Proof. Necessity. The first part of the proof is almost the same for all three assertions. Let $\mathsf{LT} \in \{\mathfrak{T}\text{-}\mathcal{REL},\ \mathcal{R}\text{-}\mathcal{REL},\ \mathcal{LIM}\}$ and let $\mathcal{U} \subseteq \mathsf{LT}(\mathsf{S})$ for some strategy $\mathsf{S} \in \mathcal{R}$. The desired operator $\mathfrak{O}$ is defined as follows. Let $\mathsf{f} \in \mathcal{M}$ and let $\mathsf{x} \in \mathbb{N}$.

$\mathfrak{O}(f,x) =$ "Compute $S(f^x)$. Use half of the time for executing (A) and (B) until (C) or (D) happens.

(A) Compute $S(f^{x+1}),\ S(f^{x+2}),\dots$

(B) Check if $\Phi_{S(f^x)}(x) = y$ for $y = 0,\ 1,\ 2,\ \dots$

(C) In (A) a $k \in \mathbb{N}$ is found such that $S(f^x) \neq S(f^{x+k})$. Set $\mathfrak{O}(f,x) = 0$.

(D) In (B) a $y \in \mathbb{N}$ is found such that $\Phi_{S(f^x)}(x) = y$. Set $\mathfrak{O}(f,x) = \Phi_{S(f^x)}(x)$."

First, we show the necessity part of Assertion (1). Clearly, the operator $\mathfrak{O}$ is recursive, since by Definition 17, for all $f \in \mathfrak{T}$ and all $x \in \mathbb{N}$ we have that $S(f^x)$ is defined. Test (B) can be effectively executed by Property (2) of a complexity measure. It remains to show that $\mathfrak{O}$ is general recursive.

Claim 1. $\mathfrak{O}(\mathfrak{T}) \subseteq \mathfrak{T}$.

Suppose that for some $f \in \mathfrak{T}$ and some $x \in \mathbb{N}$ the value $\mathfrak{O}(f,x)$ is not defined. Then, in particular, (C) cannot happen. But this means that $S(f^x) = S(f^{x+n})$ for all $n \in \mathbb{N}$. Therefore, the sequence $(S(f^m)_{m\in\mathbb{N}}$ converges to $S(f^x)$. Since S is reliable on $\mathfrak{T}$, we know that $\varphi_{S(f^x)} = f$. Consequently, $\varphi_{S(f^x)}(x)$ is defined and thus, by Property (1) of a complexity measure, $\Phi_{S(f^x)}(x)$ is defined, too. Thus, in (D) a y must be found such that $\Phi_{S(f^x)}(x) = y$, a contradiction to $\mathfrak{O}(f,x)$ undefined. This proves Claim 1.

Claim 2. $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$.

Let $f \in \mathcal{U}$ be arbitrarily fixed. Since $\mathcal{U} \in \mathcal{LIM}(S)$, the sequence $(S(f^m))_{m\in\mathbb{N}}$ converges, say to j and $\varphi_j = f$. Thus, in the definition of $\mathfrak{O}(f,x)$, Test (C) can succeed only finitely often. That is, for all but finitely many x we have $\mathfrak{O}(f,x) = \Phi_{S(f^x)}(x)$. Consequently, $f \in \mathcal{R}_{\mathfrak{O}}$. Thus Claim 2 is shown and the necessity part of Assertion (1) follows.

For showing the necessity part of Assertion (2) note that the operator $\mathfrak{O}$ is effective, too. We have to show $\mathfrak{O}(\mathcal{R}) \subseteq \mathcal{R}$ instead of Claim 1, while Claim 2 and its proof remain unchanged. This can be done *mutatis mutandis* as above.

For the necessity part of Assertion (3) we again note that the operator $\mathfrak{O}$ is effective, since by Definition 8 we know that $S(f^x)$ is defined for all $f \in \mathcal{U}$ and all $x \in \mathbb{N}$. Now we have to show that $\mathfrak{O}(\mathcal{U}) \subseteq \mathcal{R}$, while Claim 2 and its proof again remain unchanged.

Claim 3. $\mathfrak{O}(\mathcal{U}) \subseteq \mathcal{R}$.

Suppose that for some $f \in \mathcal{U}$ and some $x \in \mathbb{N}$ the value $\mathfrak{O}(f,x)$ is not defined. Then, in particular, (C) cannot happen. But this means that $S(f^x) = S(f^{x+n})$ for all $n \in \mathbb{N}$. Therefore, the sequence $(S(f^m))_{m\in\mathbb{N}}$ converges to $S(f^x)$. Since $f \in \mathcal{U}$ and $\mathcal{U} \in \mathcal{LIM}(S)$, we know that $\varphi_{S(f^x)} = f$. Consequently, $\varphi_{S(f^x)}(x)$ is defined and thus, by Property (1) of a complexity measure, $\Phi_{S(f^x)}(x)$ is defined, too. Thus, in (D) a y must be found such that $\Phi_{S(f^x)}(x) = y$, a contradiction to $\mathfrak{O}(f,x)$ undefined. This proves Claim 3.

Thus, we have shown the necessity parts of Assertions (1) through (3).

Sufficiency. Again, the first part of the proof is identical for Assertions (1) through (3). Let $\mathfrak{O}$ be an operator satisfying the relevant conditions. We define the desired strategy as a limiting recursive functional.

$S(f) =$ "Execute Stage 0:

> Stage n: Compute $c_2(n) = (i, k)$. Output i.
> Check for all $x \in \mathbb{N}$ whether or not $\Phi_i(x) \leqslant \max\{k, \mathfrak{O}(f, x)\}$ and $\varphi_i(x) = f(x)$. If this test fails for some x, stop executing Stage n and goto Stage $n+1$."

Now for showing Assertions (1) through (3) it suffices to distinguish the cases $\mathcal{M} \in \{\mathfrak{T}, \mathcal{R}, \mathcal{U}\}$ and to show that S is reliable on $\mathcal{M}$. Note that these three cases are completely reflected by the domain of the operator $\mathfrak{O}$.

Claim 1. S *is reliable on* $\mathcal{M}$.

Let $f \in \mathcal{M}$. Then we can conclude that $\mathfrak{O}(f, x)$ is defined for all $x \in \mathbb{N}$. Now suppose that $f \in \mathtt{dom}(S)$, i.e., $S(f)$ converges, say to i. Since S performs a mind change every time it enters a new stage, it follows that S enters some Stage n, where $c_2(n) = (i, k)$ and never leaves it. Thus, it verifies that $\Phi_i(x) \leqslant \max\{k, \mathfrak{O}(f, x)\}$ and $\varphi_i(x) = f(x)$ for all $x \in \mathbb{N}$. This proves Claim 1.

The following claim is identical for Assertions (1) through (3).

Claim 2. $f \in \mathcal{R}_{\mathfrak{O}}$ *implies* S *learns* f.

By the definition of $\mathcal{R}_{\mathfrak{O}}$ we know that for every $f \in \mathcal{R}_{\mathfrak{O}}$ there exist i and k such that $\varphi_i = f$ and $\Phi_i(x) \leqslant \max\{k, \mathfrak{O}(f, x)\}$ for all x. Then S can never go past Stage n, where $c_2(n) = (i, k)$. It follows that S converges, and since S is reliable, it learns f. Hence, Claim 2 is shown.

By Claims 1 and 2 the theorem follows. ∎

Further characterizations in the same style as above are possible. Wiehagen [99] showed a characterization of $\mathcal{LIM}^*$, Kinber and Zeugmann [62] characterized $\mathfrak{T}\text{-}\mathcal{REL}^a$ for every $a \in \mathbb{N} \cup \{*\}$.

Note that sometimes also a different and stronger characterization of learning types in terms of complexity is possible. The first results along this line can be found in Blum and Blum [19], who also coined the term *a-posteriori characterization*. For stating such a characterization, the notion of compression index is needed.

Definition 21 (Blum and Blum [19]). *Let* (φ, Φ) *be a complexity measure, let* $f \in \mathcal{R}$*, and let* $\mathfrak{O}$ *be a general recursive operator. Then* $i \in \mathbb{N}$ *is said to be an* $\mathfrak{O}$-compression index *of* f *if*

(1) $\varphi_i = f$,

(2) $\forall j[\varphi_j = f \rightarrow \forall x[\Phi_i(x) \leqslant \mathfrak{O}(\Phi_j, \max\{i, j, x\})]]$.

In this case we also say that the function f *is everywhere* $\mathfrak{O}$*-compressed.*

Then Blum and Blum [19] could prove the following characterization.

Theorem 36 (Blum and Blum [19]). *Let* $\mathcal{U} \subseteq \mathcal{R}$*, then we have:* $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{REL}$ *if and only if there is a general recursive operator* $\mathfrak{O}$ *such that every function in* $\mathcal{U}$ *is everywhere* $\mathfrak{O}$*-compressed.*

Consequently, function classes that are reliably identifiable on the set $\mathcal{R}$ have the property that every function of the class does possess a fastest program modulo a general recursive operator, where "fastest program modulo a general recursive operator $\mathfrak{O}$" is formalized by the notion of $\mathfrak{O}$-compression index.

Further results along this line have been achieved. We refer the interested reader to Grabowski [49] and Zeugmann [106, 109].

7. Learning and Consistency – Part II

The main goal of this section is a thorough study of the learning power of the different models of consistent learning with and without δ-delay. As we have seen above, certain additional information can help to learn the whole class of recursive functions consistently without δ-delay, i.e., $\mathcal{CONS}^+ = \wp(\mathcal{R})$ (cf. Theorem 29) – whereas we have not yet studied the exact effect of omitting additional information in $\mathcal{CONS}^\delta$-learning. So it is only natural to analyze the learning power of $\mathcal{CONS}^\delta$-models more thoroughly. We start with $\delta = 0$. The following theorem actually states that the demand to learn consistently is a severe restriction to the learning power.

Theorem 37 (Barzdin [10], Wiehagen [97]). $\mathcal{CONS} \subset \mathcal{IT}$

Proof. $\mathcal{CONS} \subseteq \mathcal{IT}$ follows from the fact that an iterative strategy can recompute all values seen so far from the hypothesis it receives as input and then take the previous values and the new value to simulate the consistent strategy. We omit the details.

For showing the remaining part, i.e., $\mathcal{IT} \setminus \mathcal{CONS} \neq \emptyset$, we use the following class $\mathcal{U} = \{f \in \mathcal{R} \mid\ f = \alpha j p,\ \alpha \in \mathbb{N}^*,\ j \geqslant 2,\ p \in \mathcal{R}_{\{0,1\}},\ \varphi_j = f\}$, where $\varphi \in \text{Göd}$.

We set $S_0(f) = S(0, f(0)) = f(0)$ and for $n \geqslant 1$ we define

$$S(k, n, m) = \begin{cases} m, & \text{if } m \geqslant 2 \\ k\ , & \text{if } k \geqslant 2 \text{ and } m < 2 \\ 0\ , & \text{otherwise.} \end{cases}$$

By construction, $S_n(f)$ is equal to the last value $f(x) \geqslant 2$ in $(f(0), \ldots, f(n))$ and 0, if no such value exists. Thus, the definition of the class $\mathcal{U}$ directly implies $\mathcal{U} \in \mathcal{IT}_\varphi(S)$.

It remains to show $\mathcal{U} \notin \mathcal{CONS}$. We need the observation that for every $\alpha \in \mathbb{N}^*$, there is an $f \in \mathcal{U}$ such that $\alpha \sqsubseteq f$. Indeed an implicit use of the fixed point theorem (cf., e.g., Smith [88]) yields that for every $\alpha \in \mathbb{N}^*$ and every $p \in \mathcal{R}_{\{0,1\}}$, there is a $j \geqslant 2$ such that $\varphi_j = \alpha j p$.

Now suppose that there is a strategy $S \in \mathcal{P}$ such that $\mathcal{U} \in \mathcal{CONS}_\varphi(S)$. The observation made directly implies $S \in \mathcal{R}$ and for every $\alpha \in \mathbb{N}^*$, $\alpha \sqsubseteq \varphi_{S(\alpha)}$. Thus, on every $\alpha \in \mathbb{N}^*$, S always computes a consistent hypothesis. Then, again by an implicit use of the fixed point theorem, let $j \geqslant 2$ be any φ-program of the function f defined as follows: $f(0) = j$, and for any $n \in \mathbb{N}$,

$$f(n+1) = \begin{cases} 0, & \text{if} \quad S(f^n 0) \neq S(f^n) \\ 1, & \text{if} \quad S(f^n 0) = S(f^n) \text{ and } S(f^n 1) \neq S(f^n) \,. \end{cases}$$

In accordance with the observation made above and the assumption that S is consistent, one immediately verifies that $S(f^n 0) \neq S(f^n)$ or $S(f^n 1) \neq S(f^n)$ for any $n \in \mathbb{N}$. Therefore the function f is everywhere defined and we have $f \in \mathcal{U}$. On the other hand, the strategy S changes its mind infinitely often when successively fed f, a contradiction to $\mathcal{U} \in \mathcal{CONS}_\varphi(S)$. ∎

As we have seen, learning in the limit is insensitive with respect to the requirement to learn exclusively with recursive strategies (cf. Theorem 16). On the other hand, consistency is a common requirement in machine learning. Therefore, it is natural to ask whether or not the power of consistent learning algorithms further decreases if one restricts the notion of learning algorithms to the set of recursive strategies. The answer to this question is provided by our next theorem.

Theorem 38 (Wiehagen and Zeugmann [104]).

$\mathcal{T}\text{-}\mathcal{CONS} \subset \mathcal{R}\text{-}\mathcal{CONS} \subset \mathcal{CONS}$.

Proof. By definition we have $\mathcal{T}\text{-}\mathcal{CONS} \subseteq \mathcal{R}\text{-}\mathcal{CONS} \subseteq \mathcal{CONS}$. Next we show that $\mathcal{U}_{sd} \in \mathcal{R}\text{-}\mathcal{CONS} \setminus \mathcal{T}\text{-}\mathcal{CONS}$. Obviously, $\mathcal{U}_{sd} \in \mathcal{R}\text{-}\mathcal{CONS}_\varphi(S)$ by the strategy $S(f^n) = f(0)$ for all $n \in \mathbb{N}$.

Now suppose that $\mathcal{U}_{sd} \in \mathcal{T}\text{-}\mathcal{CONS}$. Since $\mathcal{U}_0 \in \mathcal{T}\text{-}\mathcal{CONS}$ and $\mathcal{T}\text{-}\mathcal{CONS}$ is closed under union (cf. Theorem 34), this would directly imply that $\mathcal{U}_0 \cup \mathcal{U}_{sd} \in \mathcal{T}\text{-}\mathcal{CONS}$, a contradiction to Theorem 19. Thus $\mathcal{T}\text{-}\mathcal{CONS} \subset \mathcal{R}\text{-}\mathcal{CONS}$.

For the proof of $\mathcal{CONS} \setminus \mathcal{R}\text{-}\mathcal{CONS} \neq \emptyset$ we use a class similar to the class above, namely $\mathcal{U} = \{f \mid f \in \mathcal{R},\ \text{either } \varphi_{f(0)} = f \text{ or } \varphi_{f(1)} = f\}$. First we show that $\mathcal{U} \in \mathcal{CONS}$. The desired strategy is defined as follows. Let $f \in \mathcal{R}$ and $n \in \mathbb{N}$.

$S(f^n) =$ "Compute in parallel $\varphi_{f(0)}(x)$ and $\varphi_{f(1)}(x)$ for all $x \leqslant n$ until (A) or (B) happens.

(A) $\varphi_{f(0)}(x) = f(x)$ for all $x \leqslant n$.

(B) $\varphi_{f(1)}(x) = f(x)$ for all $x \leqslant n$.

If (A) happens first, then output $f(0)$. If (B) happens first, then output $f(1)$. If neither (A) nor (B) happens, then $S(f^n)$ is not defined."

By the definition of $\mathcal{U}$, it is obvious that $S(f^n)$ is defined for all $f \in \mathcal{U}$ and all $n \in \mathbb{N}$. Moreover, S is clearly consistent. Hence, it suffices to prove that $(S(f^n))_{n\in\mathbb{N}}$ converges for all $f \in \mathcal{U}$. But this is also an immediate consequence of the definition of $\mathcal{U}$, since either $\varphi_{f(0)} \neq f$ or $\varphi_{f(1)} \neq f$. Hence S cannot oscillate infinitely often between $f(0)$ and $f(1)$. Consequently, $\mathcal{U} \in \mathcal{CONS}_\varphi(S)$.

Next we show that $\mathcal{U} \notin \mathcal{R}\text{-}\mathcal{CONS}$. Suppose there is a strategy $S \in \mathcal{R}$ such that $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{CONS}_\varphi(S)$. Applying Smullyan's Recursion Theorem [91], we construct a function $f \in \mathcal{U}$ such that either $S(f^n) \neq S(f^{n+1})$ for all $n \in \mathbb{N}$ or $\varphi_{S(f^x)}(y) \neq f(y)$ for some $x, y \in \mathbb{N}$ with $y \leqslant x$. Since both cases yield a contradiction to the definition of $\mathcal{R}\text{-}\mathcal{CONS}$, we are done. The desired function f is defined as follows. Let h and s be two recursive functions such that for all $i, j \in \mathbb{N}$, $\varphi_{h(i,j)}(0) = \varphi_{s(i,j)}(0) = i$ and $\varphi_{h(i,j)}(1) = \varphi_{s(i,j)}(1) = j$. For any $i, j \in \mathbb{N}, x \geqslant 2$ we proceed inductively.

Suspend the definition of $\varphi_{s(i,j)}$. Try to define $\varphi_{h(i,j)}$ for more and more arguments via the following procedure.

(T) Test whether or not (A) or (B) happens (this can be effectively checked, since $S \in \mathcal{R}$):

(A) $S(\varphi^x_{h(i,j)}0) \neq S(\varphi^x_{h(i,j)})$,

(B) $S(\varphi^x_{h(i,j)}1) \neq S(\varphi^x_{h(i,j)})$.

If (A) happens, then let $\varphi_{h(i,j)}(x+1) = 0$, let $x := x+1$, and goto (T).
In case (B) happens, set $\varphi_{h(i,j)}(x+1) = 1$, let $x := x+1$, and goto (T).
If neither (A) nor (B) happens, then define $\varphi_{h(i,j)}(x') = 0$ for all $x' > x$, and goto $(*)$.

$(*)$ Set $\varphi_{s(i,j)}(n) = \varphi_{h(i,j)}(n)$ for all $n \leqslant x$, and $\varphi_{s(i,j)}(x') = 1$ for all $x' > x$.

By Smullyan's Recursion Theorem, there are numbers i and j such that $\varphi_i = \varphi_{h(i,j)}$ and $\varphi_j = \varphi_{s(i,j)}$. Now we distinguish the following cases.

Case 1. The loop in (T) is never left.

Then we directly obtain that $\varphi_i \in \mathcal{U}$, since $\varphi_j = ij$ is just a finite function while $\varphi_i \in \mathcal{R}$. Moreover, in accordance with the definition of the loop (T), on input φ_i^n the strategy S changes its mind for all $n > 0$.

Case 2. The loop in (T) is left.

Then there exists an x such that $S(\varphi^x_{h(i,j)}0) = S(\varphi^x_{h(i,j)}1)$. Hence $S(\varphi_i^{x+1}) = S(\varphi_j^{x+1})$, since $\varphi_{h(i,j)} = \varphi_i$, $\varphi_{s(i,j)} = \varphi_j$, $\varphi_i(n) = \varphi_j(n)$ for all $n \leqslant x$ by $(*)$, as well as $\varphi_i(x+1) = 0$ and $\varphi_j(x+1) = 1$. Furthermore, φ_i, $\varphi_j \in \mathcal{R}$. Since $\varphi_i(x+1) \neq \varphi_j(x+1)$, we get $\varphi_i \neq \varphi_j$. On the other hand, $\varphi_i(0) = i$ and $\varphi_j(1) = j$. Consequently, both functions φ_i and φ_j belong to $\mathcal{U}$. But $S(\varphi_i^{x+1}) = S(\varphi_j^{x+1})$ and $\varphi_i(x+1) \neq \varphi_j(x+1)$, hence S does not work consistently on input φ_i^{x+1} or φ_j^{x+1}. This contradiction completes the proof. ∎

Using the same classes as above one can show *mutatis mutandis* the following theorem.

Theorem 39. $\mathcal{T}\text{-}\mathcal{CONS}^{\delta} \subset \mathcal{R}\text{-}\mathcal{CONS}^{\delta} \subset \mathcal{CONS}^{\delta}$ *for all* $\delta \in \mathbb{N}$.

The next result provides a more subtle insight into the difference in power between $\mathcal{T}\text{-}\mathcal{CONS}$ and $\mathcal{R}\text{-}\mathcal{CONS}$. Assertion (1) has been shown by Wiehagen and Liepe [102] and Assertion (2) has been proved in Wiehagen and Zeugmann [104].

Theorem 40.

(1) $\mathcal{FIN} \# \mathcal{T}\text{-}\mathcal{CONS}$

(2) $\mathcal{FIN} \subset \mathcal{R}\text{-}\mathcal{CONS}$.

Proof. For proving Assertion (1) note that we obviously have $\mathcal{U}_{sd} \in \mathcal{FIN}$. But $\mathcal{U}_{sd} \notin \mathcal{T}\text{-}\mathcal{CONS}$ (see the proof of Theorem 38), and thus $\mathcal{FIN} \setminus \mathcal{T}\text{-}\mathcal{CONS} \neq \emptyset$. Furthermore, by Theorem 12 we have $\mathcal{U}_{(\varphi,\Phi)} \in \mathcal{T}\text{-}\mathcal{CONS}$ for every complexity measure (φ, Φ). It remains to argue that $\mathcal{U}_{(\varphi,\Phi)} \notin \mathcal{FIN}$. But this is obvious at least for complexity measures satisfying Property `ext` via Theorem 17. Thus, $\mathcal{T}\text{-}\mathcal{CONS} \setminus \mathcal{FIN} \neq \emptyset$ and Assertion (1) follows.

Next, we prove Assertion (2). Since $\mathcal{T}\text{-}\mathcal{CONS} \subseteq \mathcal{R}\text{-}\mathcal{CONS}$, $\mathcal{R}\text{-}\mathcal{CONS} \setminus \mathcal{FIN} \neq \emptyset$ is an immediate consequence of (1). The proof of $\mathcal{FIN} \subseteq \mathcal{R}\text{-}\mathcal{CONS}$ mainly relies on the decidability of the convergence of any finite learning algorithm. Let $\mathcal{U} \in \mathcal{FIN}$, and let S be any strategy witnessing $\mathcal{U} \in \mathcal{FIN}_{\varphi}(S)$. Furthermore, let $s \in \mathcal{R}$ be any function such that $\varphi_{s(\alpha)} = \alpha 0^{\infty}$ for all $\alpha \in \mathbb{N}^*$. The desired strategy $\hat{S}$ is defined as follows. Let $f \in \mathcal{R}$ and $n \in \mathbb{N}$. Then

$\hat{S}(f^n) =$ "In parallel, try to compute $S(f^0), \ldots, S(f^n)$ for precisely n steps. Let $k \geqslant 1$ be the least number such that all values $S(f^0), \ldots, S(f^k)$ turn out to be defined, and $S(f^{k-1}) = S(f^k)$.
In case this k is found, output $S(f^k)$. Otherwise, output $s(f^n)$."

It remains to show that $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{CONS}(\hat{S})$. Obviously, $\hat{S} \in \mathcal{R}$. Now let $f \in \mathcal{U}$. We have to show that $\hat{S}$ consistently learns f.

Claim 1. $\hat{S}$ learns f.

Since $f \in \mathcal{U}$, the strategy S is defined for all inputs f^n, $n \in \mathbb{N}$. Moreover, since S finitely learns f, the sequence $(S(f^n))_{n\in\mathbb{N}}$ finitely converges to a φ–program of f. Hence, $\hat{S}$ eventually has to find the least k such that $S(f^{k-1}) = S(f^k)$, and all values $S(f^0), \ldots, S(f^k)$ are defined. By the definition of $\mathcal{FIN}$, $\varphi_{S(f^k)} = f$. Hence, $\hat{S}$ learns f.

Claim 2. For all $f \in \mathcal{U}$ and $n \in \mathbb{N}$, $\hat{S}(f^n)$ is a consistent hypothesis.

As long as $\hat{S}$ outputs $s(f^n)$, it is consistent. Suppose, $\hat{S}$ outputs $S(f^k)$ for the first time. Then it has verified that $S(f^{k-1}) = S(f^k)$. Since $f \in \mathcal{U}$, and $\mathcal{U} \in \mathcal{FIN}_{\varphi}(S)$, this directly implies $\varphi_{S(f^k)} = f$. Therefore, $\hat{S}$ again outputs a consistent hypothesis. Since this hypothesis is repeated in any subsequent learning step, the claim is shown. ∎

Theorem 41. *The following statements hold for all* $\delta \in \mathbb{N}$*:*

(1) $\mathcal{T}\text{-}\mathcal{CONS}^{\delta} \subset \mathcal{T}\text{-}\mathcal{CONS}^{\delta+1} \subset \mathfrak{T}\text{-}\mathcal{REL}$,

(2) $\mathcal{NUM} \cap \wp(\mathcal{R}_{\{0,1\}}) = \mathcal{T}\text{-}\mathcal{CONS}^{\delta} \cap \wp(\mathcal{R}_{\{0,1\}}) = \mathcal{T}\text{-}\mathcal{CONS}^{\delta+1} \cap \wp(\mathcal{R}_{\{0,1\}}) = \mathfrak{T}\text{-}\mathcal{REL} \cap \wp(\mathcal{R}_{\{0,1\}})$,

(3) $\mathcal{T}\text{-}\mathcal{CONS}^{\delta} \cap \wp(\mathcal{R}_{\{0,1\}}) \subset \mathcal{R}\text{-}\mathcal{REL} \cap \wp(\mathcal{R}_{\{0,1\}})$.

Proof. We first prove Assertion (1). Let $\delta \in \mathbb{N}$ be arbitrarily fixed. Then by Definition 15 we obviously have $\mathcal{T}\text{-}\mathcal{CONS}^{\delta} \subseteq \mathcal{T}\text{-}\mathcal{CONS}^{\delta+1}$. For showing $\mathcal{T}\text{-}\mathcal{CONS}^{\delta+1} \setminus \mathcal{T}\text{-}\mathcal{CONS}^{\delta} \neq \emptyset$ we use the following class. Let (φ, Φ) be any complexity measure; we set

$$\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} = \{f \mid f \in \mathcal{R},\ \varphi_{f(0)} = f,\ \forall x[\Phi_{f(0)}(x) \leqslant f(x+\delta+1)]\}\ .$$

Claim 1. $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \in \mathcal{T}\text{-}\mathcal{CONS}^{\delta+1}$.

The desired strategy S is defined as follows. Let $g \in \mathcal{R}$ be the function defined in the sufficiency proof of Theorem 32. For all $f \in \mathcal{R}$ and all $n \in \mathbb{N}$ we set

$$S(f^n) = \begin{cases} f(0), & \text{if } n \leqslant \delta \text{ or } n > \delta \text{ and } \Phi_{f(0)}(y) \leqslant f(y+\delta+1) \\ & \text{and } \varphi_{f(0)}(y) = f(y) \text{ for all } y \leqslant n \dot{-} \delta \dot{-} 1 \\ g(f^n), & \text{otherwise.} \end{cases}$$

Now, by construction, one easily verifies $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \in \mathcal{T}\text{-}\mathcal{CONS}^{\delta+1}(S)$. This proves Claim 1.

Claim 2. $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \notin \mathcal{T}\text{-}\mathcal{CONS}^{\delta}$.

Suppose the converse. Then there must be a strategy $S \in \mathcal{R}$ such that $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \in \mathcal{T}\text{-}\mathcal{CONS}^{\delta}(S)$. We continue by constructing a function $\varphi_{i^*} \in \mathcal{U}_{\delta+1}^{(\varphi,\Phi)}$ on which S fails.

Furthermore, let $r \in \mathcal{R}$ be such that $\Phi_i = \varphi_{r(i)}$ for all $i \in \mathbb{N}$ and r is strongly monotonously increasing, i.e., $r(i) < r(i+1)$ for all $i \in \mathbb{N}$. Then $\mathtt{Val}(r)$ is recursive (cf. Rogers [86]). Choose $s \in \mathcal{R}$ such that for all $j \in \mathbb{N}$ and for all $x \leqslant \delta$ we have

$$\varphi_{s(j)}(x) = \begin{cases} i, & \text{if there is an } i \text{ with } r(i) = j \\ 0, & \text{otherwise.} \end{cases}$$

For the further definition of $\varphi_{s(j)}$ we also use $\delta + 1$ arguments in every step. For $x = 0,\ \delta+1,\ 2\delta+2,\ 3\delta+3, \ldots$ we set

$$\begin{aligned} \varphi_{s(j)}(x+\delta+1) &= \varphi_j(x) + 1 \\ &\ \ \vdots \\ \varphi_{s(j)}(x+2\delta+1) &= \varphi_j(x+\delta) + 1 \end{aligned}$$

provided $\varphi_j(x),\ \varphi_j(x+1),\dots,\varphi_j(x+\delta)$ are all defined, $\varphi_{s(j)}^{x+\delta}$ is defined and

$$S\left(\varphi_{s(j)}^{x+\delta}\right) = S\left(\langle(\varphi_{s(j)}(0),\dots,\varphi_{s(j)}(x+\delta),\varphi_j(x),\dots,\varphi_j(x+\delta))\rangle\right)$$

and

$$\begin{array}{rcl}\varphi_{s(j)}(x+\delta+1) & = & \varphi_j(x)\\ & \vdots & \\ \varphi_{s(j)}(x+2\delta+1) & = & \varphi_j(x+\delta)\end{array}$$

provided $\varphi_j(x),\ \varphi_j(x+1),\dots,\varphi_j(x+\delta)$ are all defined, $\varphi_{s(j)}^{x+\delta}$ is defined and

$$S\left(\varphi_{s(j)}^{x+\delta}\right) \neq S\left(\langle(\varphi_{s(j)}(0),\dots,\varphi_{s(j)}(x+\delta),\varphi_j(x),\dots,\varphi_j(x+\delta))\rangle\right)\ .$$

Otherwise, $\varphi_{s(j)}(x+\delta+1),\dots,\varphi_{s(j)}(x+2\delta+1)$ remain undefined.

By the fixed point theorem (cf. Rogers [86]) there exists a number i^* such that $\varphi_{s(r(i^*))} = \varphi_{i^*}$.

Next, we show that $\varphi_{i^*} \in \mathcal{U}_{\delta+1}^{(\varphi,\Phi)}$. This is done inductively. For the induction base, by construction we have $\varphi_{i^*}(0) = \cdots = \varphi_{i^*}(\delta) = i^*$. Hence, $\Phi_{i^*}(0),\dots,\Phi_{i^*}(\delta)$ are all defined, too. Therefore, we know that $\varphi_{s(r(i^*))}^{\delta}$ is defined and so either $\varphi_{s(r(i^*))}(\delta+1) = \Phi_{i^*}(0)+1,\ \dots,\ \varphi_{s(r(i^*))}(2\delta+1) = \Phi_{i^*}(\delta)+1$ provided

$$S\left(\varphi_{s(r(i^*))}^{\delta}\right) = S\left(\langle(\varphi_{s((r(i^*))}(0),\dots,\varphi_{s((r(i^*))}(\delta),\Phi_{i^*}(0),\dots,\Phi_{i^*}(\delta))\rangle\right)$$

or $\varphi_{s(r(i^*))}(\delta+1) = \Phi_{i^*}(0),\ \dots,\ \varphi_{s(r(i^*))}(2\delta+1) = \Phi_{i^*}(\delta)$ if

$$S\left(\varphi_{s(r(i^*))}^{\delta}\right) \neq S\left(\langle(\varphi_{s((r(i^*))}(0),\dots,\varphi_{s((r(i^*))}(\delta),\Phi_{i^*}(0),\dots,\Phi_{i^*}(\delta))\rangle\right)\ .$$

Note that one of these cases must happen, since otherwise S would not be $\mathcal{T}$-consistent with δ–delay.

Hence, $\Phi_{i^*}(0) \leqslant \varphi_{i^*}(\delta+1),\dots,\Phi_{i^*}(\delta) \leqslant \varphi_{i^*}(2\delta+1)$, since $\varphi_{s(r(i^*))} = \varphi_{i^*}$. So we know that $\varphi_{i^*}(\delta+1),\dots,\varphi_{i^*}(2\delta+1)$ as well as $\Phi_{i^*}(\delta+1),\dots,\Phi_{i^*}(2\delta+1)$ are all defined. This completes the induction base.

Consequently, we have the induction hypothesis that for some $x = 0, \delta+1,\ 2\delta+2,\ 3\delta+3,\dots$ the values $\varphi_{i^*}(z)$ are defined and $\Phi_{i^*}(z) \leqslant \varphi_{i^*}(z+\delta+1)$ for all $z \leqslant x+\delta$. This of course implies $\varphi_{s(r(i^*))}^{x+\delta}$ is defined, too. The induction step is done from x to $x+\delta+1$. First, we either have $\varphi_{s(r(i^*))}(x+\delta+1) = \Phi_{i^*}(x)+1,\ \dots,\ \varphi_{s(r(i^*))}(x+2\delta+1) = \Phi_{i^*}(x+\delta)+1$ provided

$$S\left(\varphi_{s(r(i^*))}^{x+\delta}\right) = S\left(\langle(\varphi_{s(r(i^*))}(0),\dots,\varphi_{s(r(i^*))}(x+\delta),\Phi_{i^*}(x),\dots,\Phi_{i^*}(x+\delta))\rangle\right)$$

or $\varphi_{s(r(i^*))}(x+\delta+1) = \Phi_{i^*}(x)$, ..., $\varphi_{s(r(i^*))}(x+2\delta+1) = \Phi_{i^*}(x+\delta)$ if

$$S\left(\varphi^{x+\delta}_{s(r(i^*))}\right) \neq S\left(\langle(\varphi_{s(r(i^*))}(0), \dots, \varphi_{s(r(i^*))}(x+\delta), \Phi_{i^*}(x), \dots, \Phi_{i^*}(x+\delta))\rangle\right) \ .$$

Note that one of these cases must happen, since otherwise S would not be $\mathcal{T}$-consistent with δ–delay.

Therefore, $\varphi_{i^*}(x+\delta+1), \dots, \varphi_{i^*}(x+2\delta+1)$ are all defined and

$$\Phi_{i^*}(x) \leqslant \varphi_{i^*}(x+\delta+1), \dots, \Phi_{i^*}(x+\delta) \leqslant \varphi_{i^*}(x+2\delta+1) \ .$$

Now we also know that $\Phi_{i^*}(x+\delta+1)$, ..., $\Phi_{i^*}(x+2\delta+1)$ are all defined. Thus, we have shown that $\varphi_{i^*} \in \mathcal{U}^{(\varphi,\Phi)}_{\delta+1}$. Finally, by construction we directly obtain that S performs infinitely many mind changes when successively fed φ_{i^*}, a contradiction to $\mathcal{U}^{(\varphi,\Phi)}_{\delta+1} \in \mathcal{T}\text{-}\mathcal{CONS}^{\delta}(S)$. This proves Claim 2.

Taking into account that, for any $f \in \mathcal{R}$, a strategy working $\mathcal{T}$-consistently with δ–delay converges when successively fed f iff it learns f, we directly get $\mathcal{T}\text{-}\mathcal{CONS}^{\delta} \subseteq \mathfrak{T}\text{-}\mathcal{REL}$ for every $\delta \in \mathbb{N}$. Now $\mathcal{T}\text{-}\mathcal{CONS}^{\delta} \subset \mathcal{T}\text{-}\mathcal{CONS}^{\delta+1} \subseteq \mathfrak{T}\text{-}\mathcal{REL}$ for all $\delta \in \mathbb{N}$ implies $\mathcal{T}\text{-}\mathcal{CONS}^{\delta} \subset \mathfrak{T}\text{-}\mathcal{REL}$ for all $\delta \in \mathbb{N}$. This proves Assertion (1).

For Assertion (2) we only have to show $\mathfrak{T}\text{-}\mathcal{REL} \cap \wp(\mathcal{R}_{\{0,1\}}) \subseteq \mathcal{NUM} \cap \wp(\mathcal{R}_{\{0,1\}})$. For that purpose let $\mathcal{U} \in \mathfrak{T}\text{-}\mathcal{REL} \cap \wp(\mathcal{R}_{\{0,1\}})$. By Theorem 35 there is a general recursive operator $\mathfrak{O}$ such that $\mathcal{U} \subseteq \mathcal{R}_{\mathfrak{O}}$, that means

$$\mathcal{U} \subseteq \{\varphi_i \mid \forall^\infty x[\Phi_i(x) \leqslant \mathfrak{O}(\varphi_i, x)]\} \cap \mathcal{R}_{\{0,1\}} \,.$$

As each general recursive operator, $\mathfrak{O}$ can be bounded by a monotone general recursive operator $\hat{\mathfrak{O}}$, i.e., $\mathfrak{O}(f,x) \leqslant \hat{\mathfrak{O}}(f,x)$ for all $f \in \mathcal{R}$ and all $x \in \mathbb{N}$, where monotonicity of $\hat{\mathfrak{O}}$ means that $\forall^\infty x[\hat{\mathfrak{O}}(f,x) \leqslant \hat{\mathfrak{O}}(g,x)]$ for all $f, g \in \mathcal{R}$ satisfying $\forall^\infty x[f(x) \leqslant g(x)]$.

In particular, for any $\varphi_i \in \mathcal{U} \subseteq \mathcal{R}_{\{0,1\}}$ we have $\forall x[\varphi_i(x) \leqslant 1]$ and therefore $\forall^\infty x[\mathfrak{O}(\varphi_i, x) \leqslant \hat{\mathfrak{O}}(1^\infty, x)]$. Consequently

$$\mathcal{U} \subseteq \{\varphi_i \mid \forall^\infty x[\Phi_i(x) \leqslant \hat{\mathfrak{O}}(1^\infty, x)]\} \cap \mathcal{R}_{\{0,1\}} \,.$$

Since $\hat{\mathfrak{O}}$ is general recursive, the function t defined by $t(x) = \hat{\mathfrak{O}}(1^\infty, x)$ for all x is recursive. Applying Theorem 4 we can conclude $\mathcal{U} \in \mathcal{NUM}$, and Assertion (2) is shown.

Finally, Assertion (3) is an immediate consequence of Assertion (2) and Theorems 2 and 3 from Stephan and Zeugmann [94] which together show that $\mathcal{U}_{mahp} \in \mathcal{R}\text{-}\mathcal{REL} \setminus \mathcal{NUM}$. Consequently, since $\mathcal{U}_{mahp} \subseteq \mathcal{R}_{\{0,1\}}$ we have $\mathcal{NUM} \cap \wp(\mathcal{R}_{\{0,1\}}) \subset \mathcal{R}\text{-}\mathcal{REL} \cap \wp(\mathcal{R}_{\{0,1\}})$. This completes the proof. ∎

In particular, we have seen that $\mathcal{U}_{mahp} \in \mathcal{R}\text{-}\mathcal{REL}$ and thus located the appropriate learning model for inferring all functions in $\mathcal{U}_{mahp}$. Does this result also extend to the class $\mathcal{U}_{ahp}$? Interestingly, now the answer depends on the underlying complexity measure. As shown in Stephan and Zeugmann [94], there are "natural" complexity measures such that $\mathcal{U}_{ahp} \notin \mathcal{BC}$. On the other hand, there are also complexity measures such that $\mathcal{U}_{ahp} \in \mathcal{LIM}$.

Together with Theorem 34 the proof of Theorem 41 allows for a nice corollary.

Corollary 42. *For all* $\delta \in \mathbb{N}$ *we have:*

(1) $\mathcal{CONS}^{\delta} \subset \mathcal{CONS}^{\delta+1}$,

(2) $\mathcal{R}\text{-}\mathcal{CONS}^{\delta} \subset \mathcal{R}\text{-}\mathcal{CONS}^{\delta+1}$.

Proof. We use $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)}$ from the proof of Theorem 41 and the class $\mathcal{U}_0$. Clearly, $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)}$, $\mathcal{U}_0 \in \mathcal{T}\text{-}\mathcal{CONS}^{\delta+1}$ and therefore, by Theorem 34 we also have $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \cup \mathcal{U}_0 \in \mathcal{T}\text{-}\mathcal{CONS}^{\delta+1}$. Consequently, $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \cup \mathcal{U}_0 \in \mathcal{R}\text{-}\mathcal{CONS}^{\delta+1}$ and $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \cup \mathcal{U}_0 \in \mathcal{CONS}^{\delta+1}$. It remains to argue that $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \cup \mathcal{U}_0 \notin \mathcal{CONS}^{\delta}$. This will suffice, since $\mathcal{R}\text{-}\mathcal{CONS}^{\delta} \subseteq \mathcal{CONS}^{\delta}$.

Suppose the converse, i.e., there is a strategy $S \in \mathcal{P}$ such that $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \cup \mathcal{U}_0 \in \mathcal{CONS}^{\delta}(S)$. By the choice of $\mathcal{U}_0$ we can then directly conclude that $S \in \mathcal{R}$ and that S has to work consistently with δ–delay on every f^n, where $f \in \mathcal{R}$ and $n \in \mathbb{N}$. But this would imply $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \cup \mathcal{U}_0 \in \mathcal{T}\text{-}\mathcal{CONS}^{\delta}(S)$, a contradiction to $\mathcal{U}_{\delta+1}^{(\varphi,\Phi)} \notin \mathcal{T}\text{-}\mathcal{CONS}^{\delta}$. ∎

A closer look at the proof above shows that we have even proved the following corollary shedding some light on the power of our notion of δ–delay.

Corollary 43. $\mathcal{T}\text{-}\mathcal{CONS}^{\delta+1} \setminus \mathcal{CONS}^{\delta} \neq \emptyset$ *for all* $\delta \in \mathbb{N}$.

The situation is comparable to Lange and Zeugmann's [68] bounded example memory learnability BEM_k of languages from positive data, where BEM_k yields an infinite hierarchy such that $\bigcup_{k\in\mathbb{N}} BEM_k$ is a proper subclass of the class of all indexed families of recursive languages that can be conservatively learned.

On the one hand, Corollary 42 shows the strength of δ–delay. On the other hand, the δ–delay cannot compensate all the learning power that is provided by the different consistency demands on the domain of the strategies.

Theorem 44. $\mathcal{R}\text{-}\mathcal{CONS} \setminus \mathcal{T}\text{-}\mathcal{CONS}^{\delta} \neq \emptyset$ *for all* $\delta \in \mathbb{N}$.

Proof. The proof can be done by using the class $\mathcal{U}_{sd}$ of self-describing functions. Obviously, $\mathcal{U}_{sd} \in \mathcal{R}\text{-}\mathcal{CONS}(S)$ as witnessed by the strategy $S(f^n) = f(0)$ for all $f \in \mathcal{R}$ and all $n \in \mathbb{N}$. Now assuming $\mathcal{U}_{sd} \in \mathcal{T}\text{-}\mathcal{CONS}^{\delta}$ for some $\delta \in \mathbb{N}$ would directly imply that $\mathcal{U}_{sd} \cup \mathcal{U}_0 \in \mathcal{T}\text{-}\mathcal{CONS}^{\delta}$ for the same δ by Theorem 34. But this is a contradiction to $\mathcal{U}_{sd} \cup \mathcal{U}_0 \notin \mathcal{BC}$ (see Theorem 19). ∎

Finally, combining Corollary 43 and Theorem 44, we get the following incomparabilities.

Corollary 45. $\mathcal{T}\text{-}\mathcal{CONS}^{\delta} \# \mathcal{CONS}^{\mu}$ *and* $\mathcal{T}\text{-}\mathcal{CONS}^{\delta} \# \mathcal{R}\text{-}\mathcal{CONS}^{\mu}$ *for all* $\delta, \mu \in \mathbb{N}$ *provided* $\delta > \mu$.

$$\begin{array}{ccccccccc}
\mathcal{T}\text{-}\mathcal{COH} & \subset & \mathcal{T}\text{-}\mathcal{COH}^1 & \subset \cdots \subset & \mathcal{T}\text{-}\mathcal{COH}^\delta & \subset & \mathcal{T}\text{-}\mathcal{COH}^{\delta+1} & \subset \cdots \subset & \mathcal{T}\text{-}\mathcal{REL} \\
\| & & \| & & \| & & \| & & \| \\
\mathcal{T}\text{-}\mathcal{CONS} & \subset & \mathcal{T}\text{-}\mathcal{CONS}^1 & \subset \cdots \subset & \mathcal{T}\text{-}\mathcal{CONS}^\delta & \subset & \mathcal{T}\text{-}\mathcal{CONS}^{\delta+1} & \subset \cdots \subset & \mathcal{T}\text{-}\mathcal{REL} \\
\cap & & \cap & & \cap & & \cap & & \cap \\
\mathcal{R}\text{-}\mathcal{COH} & \subset & \mathcal{R}\text{-}\mathcal{COH}^1 & \subset \cdots \subset & \mathcal{R}\text{-}\mathcal{COH}^\delta & \subset & \mathcal{R}\text{-}\mathcal{COH}^{\delta+1} & \subset \cdots \# & \mathcal{R}\text{-}\mathcal{REL} \\
\| & & \| & & \| & & \| & & \| \\
\mathcal{R}\text{-}\mathcal{CONS} & \subset & \mathcal{R}\text{-}\mathcal{CONS}^1 & \subset \cdots \subset & \mathcal{R}\text{-}\mathcal{CONS}^\delta & \subset & \mathcal{R}\text{-}\mathcal{CONS}^{\delta+1} & \subset \cdots \# & \mathcal{R}\text{-}\mathcal{REL} \\
\cap & & \cap & & \cap & & \cap & & \cap \\
\mathcal{COH} & \subset & \mathcal{COH}^1 & \subset \cdots \subset & \mathcal{COH}^\delta & \subset & \mathcal{COH}^{\delta+1} & \subset \cdots \subset & \mathcal{LIM} \\
\| & & \| & & \| & & \| & & \| \\
\mathcal{CONS} & \subset & \mathcal{CONS}^1 & \subset \cdots \subset & \mathcal{CONS}^\delta & \subset & \mathcal{CONS}^{\delta+1} & \subset \cdots \subset & \mathcal{LIM}
\end{array}$$

Figure 2: Hierarchies of consistent learning with δ-delay

Figure 2 below summarizes the achieved separations and equivalences of the various coherent and consistent learning models investigated in this paper.

Another interesting relaxation of the consistency demand has been proposed by Wiehagen [99] – he called it *conformity.*

Definition 22 (Wiehagen [99]). *Let* $\mathcal{U} \subseteq \mathcal{R}$*, and let* $\psi \in \mathcal{P}^2$*. The class* $\mathcal{U}$ *is called* conformly learnable in the limit with respect to ψ *if there is a strategy* $S \in \mathcal{P}$ *such that*

(1) $\mathcal{U} \in \mathcal{LIM}_\psi(S)$,

(2) [*either* $\psi_{S(f^n)}(x)\uparrow$ *or* $\psi_{S(f^n)}(x) = f(x)$] *for all* $f \in \mathcal{U}$*, all* $n \in \mathbb{N}$ *and all* $x \leqslant n$*.*

$\mathcal{CONF}_\psi(S)$*,* $\mathcal{CONF}_\psi$ *and* $\mathcal{CONF}$ *are defined analogously to the above.*

Now one can prove the following theorem in which the second proper inclusion intuitively seems to be the more surprising one.

Theorem 46 (Wiehagen [99]). $\mathcal{CONS} \subset \mathcal{CONF} \subset \mathcal{LIM}$

Proof. Clearly, we have $\mathcal{CONS} \subseteq \mathcal{CONF} \subseteq \mathcal{LIM}$. For showing $\mathcal{LIM} \setminus \mathcal{CONF} \neq \emptyset$ only one new idea is needed. We use the class from the proof of Theorem 37, i.e., $\mathcal{U} = \{f \in \mathcal{R} \mid\ f = \alpha j p,\ \alpha \in \mathbb{N}^*,\ j \geqslant 2,\ p \in \mathcal{R}_{\{0,1\}},\ \varphi_j = f\}$. Suppose $\mathcal{U} \in \mathcal{CONF}_\varphi(S)$. Then we directly get $S \in \mathcal{R}$. Using the fixed point theorem we obtain a number j which is a φ-program of the following function f.

$$\begin{aligned}
f(0) &= j\ , \\
f(n+1) &= \begin{cases} 0, & \text{if } S(f^n) \neq S(f^n 0) \\ 1, & \text{if } S(f^n) = S(f^n 0) \text{ and } S(f^n) \neq S(f^n 1)\ , \\ 1, & \text{if } S(f^n) = S(f^n 0) = S(f^n 1)\ . \end{cases}
\end{aligned}$$

By construction we obtain $f \in \mathcal{U}$. Since S is supposed to be conform, if the case $S(f^n) = S(f^n 0) = S(f^n 1)$ occurs, we see that $\varphi_{S(f^n 1)}(n+1)$ must diverge. Consequently, $\varphi_{S(f^n 1)}(n+1) \neq f$. Hence, the sequence $(S(f^n))_{n\in\mathbb{N}}$ contains infinitely many mind changes or infinitely many wrong hypotheses. Since $f \in \mathcal{U}$ we get a contradiction to $\mathcal{U} \in \mathcal{CONF}_{\varphi}(S)$, and thus $\mathcal{CONF} \subset \mathcal{LIM}$ must hold.

For the remaining part $\mathcal{CONF} \setminus \mathcal{CONS} \neq \emptyset$ we refer the reader to [99]. ▮

The second inclusion in the theorem above seems to be of great interest for the philosophy of science: it says that, when learning in the limit, strategies using conjectures which convergently contradict known data may have a strictly greater inference power than strategies whose conjectures never contradict known data convergently.

8. Characterizations in Terms of Computable Numberings

In this survey, we have already seen several characterizations of learning types in terms of computable numberings. We started with the characterization of $\mathcal{R}$-$\mathcal{TOTAL}$ (see Theorem 2), then presented a characterization for $\mathcal{TOTAL}$ (see Theorem 10), and provided also a characterization of $\mathcal{T}$-$\mathcal{CONS}^{arb}$ in terms of measurable numberings (see Theorem 12). So far all these characterizations showed that for each learning type LT considered and each class $\mathcal{U} \in \mathsf{LT}$ there is a non-Gödel numbering ψ such that $\mathcal{U}$ is learnable in the sense of LT with respect to ψ by an enumerative inference strategy.

The technical difficulty we have to overcome is provided by Lemma 3 below. For every class $\mathcal{U} \notin \mathcal{NUM}$ there are *only* hypothesis spaces ψ such that $\mathcal{U} \subseteq \mathcal{P}_{\psi}$ implies the undecidability of the halting problem with respect to ψ.

Lemma 3 (Wiehagen and Zeugmann [103]). *Let* $\mathcal{U} \subseteq \mathcal{R}$ *and let* $\mathcal{U} \notin \mathcal{NUM}$. *Then, for any numbering* $\psi \in \mathcal{P}^2$ *satisfying* $\mathcal{U} \subseteq \mathcal{P}_{\psi}$, *the halting problem with respect to* ψ *is undecidable.*

Proof. Let $\mathcal{U} \subseteq \mathcal{R}$ and $\mathcal{U} \notin \mathcal{NUM}$. Furthermore, let $\psi \in \mathcal{P}^2$ be any numbering such that $\mathcal{U} \subseteq \mathcal{P}_{\psi}$. Suppose the halting problem with respect to ψ is decidable. So there exists a function $h \in \mathcal{R}^2$ such that for all $i, x \in \mathbb{N}$, $h(i, x) = 1$ iff $\psi_i(x)$ is defined. Then define a numbering $\tilde{\psi}$ by effectively filling out the "gaps" in ψ as follows:

$$\tilde{\psi}_i(x) = \begin{cases} \psi_i(x) & , \text{ if } h(i,x) = 1 \\ 0 & , \text{ otherwise} \end{cases}$$

Obviously, for any $i \in \mathbb{N}$, if $\psi_i \in \mathcal{R}$ then $\tilde{\psi}_i = \psi_i$. Hence $\mathcal{U} \subseteq \mathcal{P}_{\tilde{\psi}}$. However, $\tilde{\psi} \in \mathcal{R}^2$ and consequently, $\mathcal{U} \subseteq \mathcal{P}_{\tilde{\psi}}$ implies $\mathcal{U} \in \mathcal{NUM}$, a contradiction. ▮

So, it is only natural to ask whether or not we can show an analogous result for *every* learning type. What we would like to present in this chapter is substantial evidence for an affirmative answer. Besides its epistemological importance, these characterizations will also provide a deeper insight into the problem what kind of properties "inference-friendly" non-Gödel numberings ψ should have in order to make $\mathcal{R}_{\psi}$ learnable.

One idea may be derived from the proof of $\mathcal{R} \in \mathcal{CONS}^{+}$ (cf. Theorem 29, Assertion (2)). Here the additional information allowed for restricting the hypothesis search to a *finite* subspace of φ. If we can limiting recursively *compute* such additional information, then we can use essentially the same proof technique. Note that this idea goes back to Barzdin and Podnieks [18].

Theorem 47 (Wiehagen [99]). *Let* $\mathcal{U} \subseteq \mathcal{R}$, *then we have:*

(1) $\mathcal{U} \in \mathcal{LIM}$ *if and only if there exists a limiting recursive functional* B *such that* $\mathcal{U} \subseteq \mathtt{dom}(B)$ *and* $B(f) \geqslant \min_\varphi f$ *for all* $f \in \mathcal{U}$.

(2) $\mathcal{U} \in \mathcal{CONS}$ *if and only if there exists a function* $B \in \mathcal{P}$ *such that for every* $f \in \mathcal{U}$ *the following conditions are satisfied:*

 (A) *There is a* $j \geqslant \min_\varphi f$ *with* $B(f^n) = j$ *for all but finitely many* $n \in \mathbb{N}$.

 (B) $B(f^n)$ *is defined for every* $n \in \mathbb{N}$ *and there is an* $i \leqslant B(f^n)$ *such that* $\varphi_i =_n f$.

In the second version we cannot restrict the hypothesis search to a *finite* subspace of φ. Instead, the classes to learn can be embedded into a computable numbering ψ such that for computing the actual hypothesis quite often only finitely many elements of the computable numbering ψ have to be considered. Note that we have already provided a theorem that uses precisely this idea, i.e., the characterization of $\mathcal{TOTAL}$ (cf. Theorem 10). Using similar ideas Wiehagen [99] could show the following theorem.

Theorem 48 (Wiehagen [99]). *Let* $\mathcal{U} \subseteq \mathcal{R}$, *then we have:*

(1) $\mathcal{U} \in \mathcal{LIM}$ *if and only if there exists a numbering* $\psi \in \mathcal{P}^2$ *such that the following conditions are satisfied:*

 (A) $\mathcal{U} \subseteq \mathcal{P}_\psi$

 (B) *There is a function* $g \in \mathcal{R}$ *such that for every function* $f \in \mathcal{U}$ *the set of all numbers* i *with* $\psi_i =_{g(i)} f$ *is finite.*

(2) $\mathcal{U} \in \mathcal{BC}$ *if and only if there exists a numbering* $\psi \in \mathcal{P}^2$ *such that the following conditions are satisfied:*

 (A) $\mathcal{U} \subseteq \mathcal{P}_\psi$

 (B) *There is a function* $r \in \mathcal{R}$ *such that for every function* $f \in \mathcal{U}$ *and almost all* i, $\psi_i =_{r(i)} f$ *implies* $\psi_i = f$.

Proof. We only prove Assertion (2) here, since Assertion (1) can be shown *mutatis mutandis* as Theorem 10.

In order to show the necessity part of (2) let $\mathcal{U} \in \mathcal{BC}_\varphi(S)$, where, without loss of generality, $S \in \mathcal{R}$ (cf. Theorem 22). We define M to be the set of all pairs (z, n) such that

- for all $x \leqslant n$, $\varphi_z(x)$ is defined, and

- $S(\varphi_z^n) = z$.

Let M be enumerated without repetitions by $d \in \mathcal{R}$. For $i, x \in \mathbb{N}$, $d(i) = (z, n)$, we define $\psi(i, x) = \varphi_z(x)$ and $r(i) = n$.

$\mathcal{U} \subseteq \mathcal{P}_\psi$ is obvious. Let $f \in \mathcal{U}$ and let $\hat{n}$ be such that $\varphi_{S(f^n)} = f$ for all $n \geqslant \hat{n}$. Then $\psi_i =_{r(i)} f$ and $r(i) \geqslant \hat{n}$ implies $\psi_i = f$. Since there are only finitely many numbers i such that $\psi_i =_{r(i)} f$ and $r(i) < \hat{n}$, Condition (B) follows.

For showing the sufficiency part of (2) we have to define a strategy S such that $\mathcal{U} \in \mathcal{BC}_\varphi(S)$. Let amal be the amalgamation function defined in the proof of Theorem 29. Furthermore, let $c \in \mathcal{R}$ be a compiler function such that $\psi_i = \varphi_{c(i)}$ for all $i \in \mathbb{N}$. For any input f^n we define the set

$$\begin{aligned} M(f^n) \;=\; & \{i \mid i \in \mathbb{N},\ i \leqslant n,\ r(i) \leqslant n \\ & \wedge \forall x[x \leqslant r(i) \rightarrow \Phi_{c(i)}(x) \leqslant n \wedge \psi_i =_{r(i)} f] \\ & \wedge \forall x[n \geqslant x > r(i) \wedge \Phi_{c(i)}(x) \leqslant n \rightarrow \psi_i(x) = f(x)]\}\ , \end{aligned}$$

where $r \in \mathcal{R}$ is the function from Condition (B). Clearly, $M(f^n)$ is finite and computable for every f^n. Again, we choose $g \in \mathcal{R}$ such that $\varphi_{g(\langle\alpha\rangle)}(x) = y_x$ for every tuple $\alpha \in \mathbb{N}^*$, $\alpha = (y_0, \dots, y_n)$ and all $x \leqslant n$.

We define the strategy S as follows. If $M(f^n) = \emptyset$ then we set $S(f^n) = g(f^n)$. If $M(f^n) \neq \emptyset$, we set $S(f^n) = \text{amal}(\{c(i) \mid i \in M(f^n)\})$.

It remains to show that S learns every function $f \in \mathcal{U}$ behaviorally correctly. By construction and Condition (B) we know that for almost all n the set $M(f^n)$ contains only ψ-programs i such that $\psi_i \subseteq f$. For sufficiently large n, we also know by Condition (A) that $M(f^n)$ contains at least one ψ-program i such that $\psi_i = f$. Consequently, for all sufficiently large n, $S(f^n)$ is a φ-program for f, and thus $\mathcal{U} \in \mathcal{BC}_\varphi(S)$. Note that S outputs infinitely many different φ-programs for f if there are infinitely many ψ-programs for f. ∎

The third version of characterizations in terms of computable numberings shows that learnable functions classes are embeddable into numberings ψ possessing some effective distinguishability property. Informally, ψ has an effective distinguishability property if there is an effective method to distinguish ψ_i and ψ_j for every i, j provided $\psi_i \neq \psi_j$.

The following characterization is due to Wiehagen [100].

Theorem 49 (Wiehagen [99, 100]). *Let $\mathcal{U} \subseteq \mathcal{R}$, then we have: $\mathcal{U} \in \mathcal{LIM}$ if and only if there exists a numbering $\psi \in \mathcal{P}^2$ such that*

(1) *$\mathcal{U} \subseteq \mathcal{P}_\psi$, and*

(2) *there is a function $d \in \mathcal{R}^2$ such that $\psi_i \neq_{d(i,j)} \psi_j$ for all $i, j \in \mathbb{N}$ with $i \neq j$.*

Proof. Necessity. Let $\mathcal{U} \in \mathcal{LIM}$; then there exists a numbering $\varphi \in \mathcal{P}^2$ and a strategy $S \in \mathcal{P}$ such that $\mathcal{U} \in \mathcal{LIM}_\varphi(S)$.

Let M denote the set of all pairs (z, n) such that

- for all $x \leqslant n$, $\varphi_z(x)$ is defined, and
- $S(\varphi_z^{n-1}) \neq S(\varphi_z^n) = z$.

Intuitively, M corresponds to the set of all initial segments φ_z^n on which after a (perhaps last) mind change (namely $S(\varphi_z^{n-1}) \neq S(\varphi_z^n)$) the strategy S outputs a reasonable hypothesis.

Clearly, M is recursively enumerable. Let M be enumerated by $e \in \mathcal{R}$ without repetition. Now we are ready to define the desired numbering ψ. For any i such that $e(i) = (z, n)$, we set:

$$\psi_i(x) = \begin{cases} \varphi_z(x), & \text{if } x \leqslant n \\ \varphi_z(x), & \text{if } x > n \text{ and for every } y \text{ with } n < y \leqslant x, \\ & \varphi_z(y) \text{ is defined and } S(\varphi_z^y) = z \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Next, let $g \in \mathcal{R}$ be chosen such that $g(i) = n$ for every i with $e(i) = (z, n)$. We define $d(i, j) = \max\{g(i), g(j)\}$.

It remains to show that Conditions (1) and (2) are satisfied.

Claim 1. $\mathcal{U} \subseteq \mathcal{P}_\psi$.

Let $f \in \mathcal{U}$; we have to show that there is an i such that $\psi_i = f$. Since $f \in \mathcal{U}$, there exists a least n such that $S(f^n) = S(f^{n+m}) = z$ for all $m \in \mathbb{N}$. Then $(z, n) \in M$ and since S has converged and since $\mathcal{U} \in \mathcal{LIM}_\varphi(S)$, we also have $\psi_i = f$. Thus, Claim 1 is shown.

Claim 2. $\psi_i \neq_{d(i,j)} \psi_j$ *for all* $i, j \in \mathbb{N}$ *with* $i \neq j$.

Let $i \neq j$ and suppose that $\psi_i =_{d(i,j)} \psi_j$. Without loss of generality we can assume $g(i) < g(j)$. By the definition of M it follows that $\psi_j(x)$ is defined for all $x \leqslant g(j)$ and thus $\psi_i(x)$ is defined for all $x \leqslant g(j)$, too. By the definition of ψ we obtain that $S(\psi_i^{g(j)-1}) = S(\psi_i^{g(j)})$. On the other hand, by the definition of M we get that $S(\psi_j^{g(j)-1}) \neq S(\psi_j^{g(j)})$. But this is a contradiction to $\psi_i =_{d(i,j)} \psi_j$. Thus, Claim 2 is proved and the necessity part of the theorem follows.

Sufficiency. We define the desired strategy as follows.

$S(f^0) = 0$

$S(f^n) =$ "Compute $i = S(f^{n-1})$. Check within at most n steps of computation whether or not there is a $j > i$ such that $d(i, j) \leqslant n$ and $\psi_i =_{d(i,j)} f$. If such a j is found, output $i + 1$.
Otherwise output i."

We have to show that $\mathcal{U} \in \mathcal{LIM}_{\psi}(S)$. Let $f \in \mathcal{U}$, let $n \in \mathbb{N}^{+}$ and let $i = S(f^{n-1})$. Suppose there is a $j > i$ such that $d(i,j) \leqslant n$ and $\psi_i =_{d(i,j)} f$. Then, by Condition (2), we know that $\psi_j \neq_{d(i,j)} \psi_i$. Consequently, $\psi_i \neq f$. Thus, choosing a new hypothesis is justified. Moreover, this observation also shows that the strategy S will never abandon a correct hypothesis i.

So, it remains to show that i is abandoned if $\psi_i \neq f$. Fix any i with $\psi_i \neq f$. Then, by Condition (1), there is a $j > i$ (namely $\psi_j = f$) such that $d(i,j) \leqslant n$, for n large enough, and $\psi_j =_{d(i,j)} f$. Thus, j will be eventually found and the strategy is forced to change the provably wrong hypothesis i to $i+1$.

Putting it all together, we get that for every $f \in \mathcal{U}$ the strategy S converges to the minimal (and only!) ψ–number of f. This shows $\mathcal{U} \in \mathcal{LIM}_{\psi}(S)$. ∎

This theorem nicely shows that requiring a learning strategy to exclusively output programs for recursive functions is by no means the only way to realize Popper's [84] refutability principle. Instead, Theorem 49 leads to the crucial notion of *semantic finiteness*. Intuitively, a semantically finite strategy is never allowed to reject a hypothesis that is correct for the target function. Hence, when learning semantically finitely, a strategy should have a serious reason to reject its current hypothesis. As the proof of Theorem 49 shows, this reason might be quite different from just detecting an inconsistency.

Next, we turn our attention to finite learning. As we have already seen, no class $\mathcal{U} \in \mathcal{FIN}$ can contain an accumulation point (cf. Theorem 17). A closer look at this observation leads to the following characterization of $\mathcal{FIN}$.

Theorem 50 (Wiehagen [98, 100]). *Let $\mathcal{U} \subseteq \mathcal{R}$, then we have: $\mathcal{U} \in \mathcal{FIN}$ if and only if there exists a numbering $\psi \in \mathcal{P}^2$ such that*

(1) $\mathcal{U} \subseteq \mathcal{P}_{\psi}$, *and*

(2) *there is a function $d \in \mathcal{R}$ such that $\psi_i \neq_{d(i)} \psi_j$ for all $i, j \in \mathbb{N}$ with $i \neq j$.*

Proof. Necessity. Let $\mathcal{U} \in \mathcal{FIN}$; then there exists a numbering $\varphi \in \mathcal{P}^2$ and a strategy $S \in \mathcal{P}$ such that $\mathcal{U} \in \mathcal{FIN}_{\varphi}(S)$.

Let M denote the set of all pairs (z, n) such that

- for all $x \leqslant n$, $\varphi_z(x)$ is defined,
- for all $0 < x < n$, $S(\varphi_z^{x-1}) \neq S(\varphi_z^{x})$, and
- $S(\varphi_z^{n-1}) = S(\varphi_z^{n}) = z$.

Let M be enumerated by $e \in \mathcal{R}$ without repetition. For any i such that $e(i) = (z, n)$ we define:

$$\psi_i = \varphi_z \text{ and } d(i) = n\ .$$

Clearly, $\psi \in \mathcal{P}^2$ and $d \in \mathcal{R}$. It remains to show that Conditions (1) and (2) are satisfied.

Claim 1. $\mathcal{U} \subseteq \mathcal{P}_\psi$.

Let $f \in \mathcal{U}$; we know that $\{f\} \in \mathcal{FIN}_\varphi(S)$. Thus, there exists an $n \in \mathbb{N}$ such that $S(f^{x-1}) \neq S(f^x)$ for all $0 < x < n$ and $S(f^{n-1}) = S(f^n) = z$. By the definition of finite convergence, we can conclude that $(S(f^n))_{n\in\mathbb{N}}$ has converged and since $\{f\} \in \mathcal{FIN}_\varphi(S)$, we know that $\varphi_z = f$. Thus, $(z, n) \in M$ and by construction there is an i such that $e(i) = (z, n)$. Hence, $\psi_i = \varphi_z = f$. This proves Claim 1.

Claim 2. $\psi_i \neq_{d(i)} \psi_j$ *for all* $i, j \in \mathbb{N}$ *with* $i \neq j$.

Let $i \neq j$ and suppose $\psi_i =_{d(i)} \psi_j$. Let $e(i) = (z, n)$; by the definition of d we can conclude $\psi_i(x)\downarrow$ for all $x = 0, \dots, n$. Additionally, by the definition of the relation $=_m$ we also have that $\psi_j(x)\downarrow$ for all $x = 0, \dots, n$. Now let $e(j) = (\hat{z}, \hat{n})$. Since $i \neq j$ and since e enumerates M without repetition, it must hold $z \neq \hat{z}$ or $n \neq \hat{n}$.

By construction, $S(\psi_i^n) = z$ and since $\psi_i =_{d(i)} \psi_j$, we also have $S(\psi_j^n) = z$. Thus, $z \neq \hat{z}$ cannot happen. But $n \neq \hat{n}$ cannot happen either, since n is the least number such that $S(\psi_i^{n-1}) = S(\psi_i^n)$. Thus $i = j$, a contradiction.

Claim 1 and 2 together yield the necessity part.

Sufficiency. The desired strategy S is defined as follows. Let $f \in \mathcal{U}$ and let $n \in \mathbb{N}$. We set:

$S(f^n) =$ "Check whether there is an $i \in \mathbb{N}$ such that

- $i \leqslant n$,
- $d(i) \leqslant n$,
- $\psi_i =_{d(i)} f$ can be verified within n steps of computation.

If such an i is found, let $S(f^n) = i$.
Otherwise, let $S(f^n) = n$."

Let $f \in \mathcal{U}$; then Condition (1) ensures that there is at least one ψ-program i such that $\psi_i =_{d(i)} f$. Furthermore, Condition (2) guarantees that there is at most one such ψ-program. Since n increases, this ψ-program i will be found eventually. Consequently, $\mathcal{U} \in \mathcal{FIN}_\psi(S)$ and thus $\mathcal{U} \in \mathcal{FIN}$. ∎

Next, we characterize the different versions of consistent learning in terms of computable numberings. As we shall see, the difference between the different versions of consistent learning can be completely expressed by different versions of consistency-related decision problems. Therefore, following Wiehagen and Zeugmann [104], next we define these consistency-related decision problems.

Definition 23. *Let* $\psi \in \mathcal{P}^2$ *be any numbering and let* $\mathcal{U} \subseteq \mathcal{R}$. *We say that*

(1) consistency with respect to ψ is decidable *if there is a predicate cons* $\in \mathcal{R}^2$ *such that for each* $\alpha \in \mathbb{N}^*$ *and all* $i \in \mathbb{N}$, *cons*$(\langle\alpha\rangle, i) = 1$ *if and only if* $\alpha \sqsubseteq \psi_i$.

(2) $\mathcal{U}$–consistency with respect to ψ is decidable *if there is a predicate cons* $\in \mathcal{P}^2$ *such that for each* $\alpha \in [\mathcal{U}]$ *and all* $i \in \mathbb{N}$, *cons*$(\langle\alpha\rangle, i)$ *is defined, and cons*$(\langle\alpha\rangle, i) = 1$ *if and only if* $\alpha \sqsubseteq \psi_i$.

(3) $\mathcal{U}$–consistency with respect to ψ is $\mathcal{R}$–decidable *if there is a predicate cons* $\in \mathcal{R}^2$ *such that for each* $\alpha \in [\mathcal{U}]$ *and all* $i \in \mathbb{N}$, *cons*$(\langle\alpha\rangle, i) = 1$ *if and only if* $\alpha \sqsubseteq \psi_i$.

Note that the following proof uses ideas from Wiehagen [98] as well as from [104].

Theorem 51. $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}$ *iff there is a numbering* $\psi \in \mathcal{P}^2$ *such that*

(1) $\mathcal{U} \subseteq \mathcal{P}_\psi$,

(2) *consistency with respect to* ψ *is decidable.*

Proof. Necessity. Let $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}_\varphi(S)$ where $\varphi \in \mathcal{P}^2$ is any Gödel numbering and S is a $\mathcal{T}$–consistent strategy. Let

$$M = \{(z, n) \mid z, n \in \mathbb{N},\ \varphi_z(x) \text{ is defined for every } x \leqslant n,\ S(\varphi_z^n) = z\}$$

be recursively enumerated by a function e. Then define a numbering ψ as follows. Let $i, x \in \mathbb{N}$, $e(i) = (z, n)$.

$$\psi_i(x) = \begin{cases} \varphi_z(x), & \text{if } x \leqslant n \\ \varphi_z(x), & \text{if } x > n \text{ and, for any } y \in \mathbb{N} \text{ such that } n < y \leqslant x, \\ & \varphi_z(y) \text{ is defined and } S(\varphi_z^y) = z \\ \uparrow, & \text{otherwise.} \end{cases}$$

For showing (1) let $f \in \mathcal{U}$ and $n, z \in \mathbb{N}$ be such that $S(f^m) = z$ for any $m \geqslant n$. Clearly, $\varphi_z = f$. Furthermore, $(z, n) \in M$. Let $i \in \mathbb{N}$ be such that $e(i) = (z, n)$. Then, by definition of ψ, $\psi_i = \varphi_z = f$. Hence $\mathcal{U} \subseteq \mathcal{P}_\psi$.

In order to prove (2) we define *cons* $\in \mathcal{R}^2$ such that for any $\alpha \in \mathbb{N}^*$, $i \in \mathbb{N}$, *cons*$(\langle\alpha\rangle, i) = 1$ iff $\alpha \sqsubseteq \psi_i$. Let $\alpha = (\alpha_0, \ldots, \alpha_x) \in \mathbb{N}^*$ and $i \in \mathbb{N}$. Let $e(i) = (z, n)$. Then define

$$\textit{cons}(\langle\alpha\rangle, i) = \begin{cases} 1, & \text{if } x \leqslant n \text{ and, for every } y \leqslant x,\ \alpha_y = \psi_i(y) \\ 1, & \text{if } x > n \text{ and } S(\langle\alpha_0, \ldots, \alpha_y\rangle) = z \text{ for every } y \in \mathbb{N} \\ & \text{such that } n < y \leqslant x \\ 0, & \text{otherwise.} \end{cases}$$

Since $e(i) = (z, n) \in M$, by construction we know that $\varphi_z(m)\downarrow$ for all $m \leqslant n$ and $S(\varphi_z^n) = z$. Thus, we have $\psi_i(m) = \varphi_z(m)$ for all $m \leqslant n$. Consequently, if $x \leqslant n$, then for all $y \leqslant x$ it can be effectively tested whether or not $\alpha_y = \psi_i(y)$. Furthermore, $S \in \mathcal{R}$ implies that $S(\langle\alpha_0, \ldots, \alpha_y\rangle)$ can be computed for every $y \in \mathbb{N}$ such that $n < y \leqslant x$. Thus, if $x > n$, the condition $S(\langle\alpha_0, \ldots, \alpha_y\rangle) = z$ can be effectively checked for every $y \in \mathbb{N}$ such that $n < y \leqslant x$. Consequently, *cons* $\in \mathcal{R}^2$.

Furthermore, it is not hard to see that for every $\alpha \in \mathbb{N}^*$, $i \in \mathbb{N}$, we have $cons(\langle\alpha\rangle, i) = 1$ iff $\alpha \sqsubseteq \psi_i$. This proves the necessity part.

Sufficiency. Let $\psi \in \mathcal{P}^2$ be any numbering. Let $cons \in \mathcal{R}^2$ be such that for all $\alpha \in \mathbb{N}^*$ and $i \in \mathbb{N}$ we have $cons(\alpha, i) = 1$ iff $\alpha \sqsubseteq \psi_i$. Let $\mathcal{U} \subseteq \mathcal{P}_\psi$. In order to consistently learn any function $f \in \mathcal{U}$ it suffices to define $S(f^n) = \min\{i \mid cons(f^n, i)\}$. However, S is undefined if, for $f \notin \mathcal{U}$, $n \in \mathbb{N}$, there is *no* $i \in \mathbb{N}$ such that $f^n \sqsubseteq \psi_i$. The following more careful definition of S will circumvent this difficulty. Let $\varphi \in$ Göd. Let $aux \in \mathcal{R}$ be such that for any $\alpha \in \mathbb{N}^*$, $\varphi_{aux(ff)} = \alpha 0^\infty$. Finally, let $c \in \mathcal{R}$ be such that for all $i \in \mathbb{N}$, $\psi_i = \varphi_{c(i)}$. Then, for any $f \in \mathcal{R}$, $n \in \mathbb{N}$, define a strategy S as follows.

$$S(f^n) = \begin{cases} c(j), & \text{if } I = \{i \mid i \leqslant n,\ cons(f^n, i) = 1\} \neq \emptyset \text{ and } j = \min\ I \\ aux(f^n), & \text{if } I = \emptyset. \end{cases}$$

Clearly, $S \in \mathcal{R}$ and S outputs only consistent hypotheses. Now let $f \in \mathcal{U}$. Then, obviously, $(S(f^n))_{n\in\mathbb{N}}$ converges to $c(\min\{i \mid \psi_i = f\})$. Hence, the strategy S witnesses $\mathcal{U} \in \mathcal{T}\text{-}\mathcal{CONS}_\varphi$. ∎

Next, we present our characterization for $\mathcal{CONS}$.

Theorem 52. $\mathcal{U} \in \mathcal{CONS}$ *iff there is a numbering* $\psi \in \mathcal{P}^2$ *such that*

(1) $\mathcal{U} \subseteq \mathcal{P}_\psi$,

(2) $\mathcal{U}$*–consistency with respect to* ψ *is decidable.*

Finally, we characterize $\mathcal{R}\text{-}\mathcal{CONS}$.

Theorem 53. $\mathcal{U} \in \mathcal{R}\text{-}\mathcal{CONS}$ *iff there is a numbering* $\psi \in \mathcal{P}^2$ *such that*

(1) $\mathcal{U} \subseteq \mathcal{P}_\psi$,

(2) $\mathcal{U}$*–consistency with respect to* ψ *is* $\mathcal{R}$*–decidable.*

The proofs of Theorems 52 and 53 are similar to that of Theorem 51.

The above characterizations of $\mathcal{T}\text{-}\mathcal{CONS}$, $\mathcal{CONS}$ and $\mathcal{R}\text{-}\mathcal{CONS}$ as well as the characterization of $\mathcal{T}\text{-}\mathcal{CONS}^{arb}$ provided in Theorem 12 point out a relation between the problem of deciding consistency and the halting problem. On the one hand, for any of the learning types $LT \in \{\mathcal{T}\text{-}\mathcal{CONS},\ \mathcal{CONS},\ \mathcal{R}\text{-}\mathcal{CONS},$
$\mathcal{T}\text{-}\mathcal{CONS}^{arb}\}$ we have $\mathcal{NUM} \subseteq LT$ (via Theorem 2 and Corollary 13). On the other hand, as shown in Lemma 3, for any class $\mathcal{U} \subseteq \mathcal{R}$ outside $\mathcal{NUM}$ and any numbering $\psi \in \mathcal{P}^2$, if $\mathcal{U} \subseteq \mathcal{P}_\psi$, then the halting problem with respect to ψ is undecidable.

In contrast, for any $\mathcal{U} \in LT \setminus \mathcal{NUM}$ the corresponding version of consistency with respect to ψ is decidable. Hence this version of consistency *cannot* be decided by first deciding the halting problem and second, if possible, computing the desired values of the function under consideration in order to compare these values with the given ones. So, though consistency is decidable in the "characteristic" numberings of Theorems 51, 52, 53 and 12 it is not decidable in a "straightforward way."

9. Further Topics

In this section we briefly summarize further research pursued by Rolf Wiehagen and his co-workers. We start with robust inference.

9.1. *Robust Learning*

For the sake of motivation, let us look at any class $\mathcal{U}$ in $\mathcal{R}\text{-}\mathcal{TOTAL}$. Recalling that $\mathcal{R}\text{-}\mathcal{TOTAL} = \mathcal{NUM}$ we can think of $\mathcal{U}$ as (a subset of) a family $(\varphi_{s(i)})_{i\in\mathbb{N}}$ for some $s \in \mathcal{R}$ (cf. Theorem 2). Furthermore, let $\mathfrak{O}$ be any effective operator realized by a function $g \in \mathcal{R}$ such that $\mathfrak{O}(\mathcal{U}) \subseteq \mathcal{R}$ (cf. Definition 20). Then we have $\mathfrak{O}(\varphi_{s(i)}) = \varphi_{g(s(i))}$ for every $i \in \mathbb{N}$ and thus the family $(\varphi_{g(s(i))})_{i\in\mathbb{N}}$ also belongs to $\mathcal{NUM}$ and is consequently in $\mathcal{R}\text{-}\mathcal{TOTAL}$, again by Theorem 2.

Thus, every class $\mathcal{U}$ in $\mathcal{R}\text{-}\mathcal{TOTAL}$ has the remarkable property that not only the class $\mathcal{U}$ itself is learnable but all classes obtained from $\mathcal{U}$ by effective transformations are $\mathcal{R}\text{-}\mathcal{TOTAL}$-learnable, too. This property may serve as a first informal definition of what is meant by saying that a class is *robustly learnable.*

For discussing the importance of robust learnability, it is helpful to recall that we have separated several learning types by using some class of self-describing functions, for example $\mathcal{U}_{sd}$. On the one hand, this is an elegant proof method and, as pointed out by Jain, Smith and Wiehagen [57], self-description is quite a natural phenomenon in that every cell of every organism contains a description of itself. On the other hand, such separating classes may seem a bit artificial, since they use coding tricks. So for the positive part of the separation, a learner only needs to fetch some code from the input. For the non-learnability result one usually shows that at least one function in the separating class is too complex to gain the information necessary to learn it in the more restricted model. If such self-describing function classes were the only separating examples, then we would have to draw major consequences for our overall understanding of learning and for the value of the theory developed so far.

Around thirty years ago, Bārzdiņš suggested to prove or to disprove the following conjecture.

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

An affirmative answer to Bārzdiņš' conjecture could be interpreted as follows. Every function class $\mathcal{U}$ in $\mathcal{LIM} \setminus \mathcal{NUM}$ contains only functions having encoded a certain information in their graphs which is helpful in the learning process. Intuitively, this information is then erased by some operator $\mathfrak{O}$ and thus $\mathfrak{O}(\mathcal{U}) \notin \mathcal{LIM}$.

However, it took many years before any progress concerning Bārzdiņš' conjecture has been made. Zeugmann [108] proposed to generalize Bārzdiņš' conjecture by replacing $\mathcal{LIM}$ by any learning type LT and showed it to be true for $\mathcal{FIN}$ and $\mathcal{T}$-$\mathcal{REL}$. Then Kurtz and Smith [64, 65] disproved Bārzdiņš' conjecture for classes $\mathcal{U} \in \mathcal{NUM}$. The major breakthrough has been made by Fulk [44] who also coined the term of *robust learnability*. Let LT be any learning type. Then we call a class $\mathcal{U}$ robustly LT-learnable, iff, for every operator $\mathfrak{O}$, the class $\mathfrak{O}(\mathcal{U})$ is LT-learnable.

There were many discussions about which operators $\mathfrak{O}$ are admissible in this context. Fulk [44] considered the class of all general recursive operators and for this version he disproved Bārzdiņš' conjecture for $\mathcal{LIM}$. Subsequently, many interesting results have been obtained in this context (cf., e.g., [4, 25, 53, 57, 80, 94]).

We would like to mention here only some of the results obtained. For this purpose we use LT-robust to denote the family of all classes $\mathcal{U} \subseteq \mathcal{R}$ such that $\mathfrak{O}(\mathcal{U}) \in \mathsf{LT}$ for every general recursive operator $\mathfrak{O}$.

As we have seen in Theorem 21, there is an infinite anomaly hierarchy for $\mathcal{LIM}$-type learning and also for $\mathcal{BC}$-type identification (cf. Case and Smith [27]). These hierarchies do *not* stand robustly.

Theorem 54 (Fulk [44]).

(1) $\mathcal{LIM}^a$*-robust* $= \mathcal{LIM}$*-robust for all* $a \in \mathbb{N} \cup \{*\}$.

(2) $\mathcal{BC}^a$*-robust* $= \mathcal{BC}$*-robust for all* $a \in \mathbb{N} \cup \{*\}$.

Let us denote by $\mathcal{LIM}_n$ the family of all classes $\mathcal{U} \subseteq \mathcal{R}$ which can be learned in the limit with at most n mind changes. Then it is not hard to prove that $\mathcal{LIM}_n \subset \mathcal{LIM}_{n+1}$ for every $n \in \mathbb{N}$ (cf. [27]).

Interestingly, this mind change hierarchy *does stand* robustly as the following theorem shows. Note that the proof uses a complicated priority argument.

Theorem 55 (Jain, Smith and Wiehagen [57]).
$\mathcal{LIM}_n$*-robust* $\subset \mathcal{LIM}_{n+1}$*-robust for all* $n \in \mathbb{N}$.

In [57] it was also shown that the $\mathcal{LIM}_{team}(n)$ and $\mathcal{BC}_{team}(n)$ hierarchies stand robustly.

A more detailed study on which learning types contain rich robustly learnable classes has been carried out by Case *et al.* [26] and the reader is encouraged to consult this paper for many deep and interesting results. As a final example, we mention the following. Adding a uniformity condition (see also Subsection 9.4) in [26] it could be proved that $\mathcal{LIM}$-uniform-robust $\subseteq \mathcal{CONS}$. This for sure sheds additional light on the importance of consistent learning.

Before closing this subsection, let us again take a look at our five classes from Section 3. Since $\mathcal{U}_0 \in \mathcal{NUM}$, it is robustly learnable. $\mathcal{U}_{sd}$ is *not* robustly learnable, since for $\mathfrak{O}(f) = g$, where $g(x) = f(x+1)$ we have $\mathfrak{O}(\mathcal{U}_{sd}) = \mathcal{R}$. Likewise, in general the class $\mathcal{U}_{(\varphi,\Phi)}$ is not in $\mathcal{LIM}$-robust. This can be seen by using Theorem 2.4 and Corollary 2.6 in [80]. As far as $\mathcal{U}_{mahp}$ and $\mathcal{U}_{ahp}$ are concerned, now the answer depends for *both* classes on the underlying complexity measure. There are complexity measures such that $\mathcal{U}_{mahp}, \mathcal{U}_{ahp} \in \mathcal{LIM}$-robust (cf. Theorem 16 and Corollary 17 in [94]). Furthermore, there are "natural" complexity measures such that $\mathcal{U}_{mahp} \notin \mathcal{LIM}$-robust (cf. Theorem 19 in [94]).

Next, we turn our attention to learning from *good examples*, motivated partly by the intuitive thought that humans can learn more efficiently from well-chosen (good) examples than they can from arbitrary input.

9.2. *Assisting the Learner*

The most natural way to help a learner – at least when thinking of the way humans learn – would be to emphasize particularly representative or helpful examples during the learning process and maybe not to present unhelpful examples at all. In particular, thus one could think of learning from only finitely many examples instead of learning from a whole infinite sequence of examples representing a target function. On a formal level, this requires a notion of what helpful examples, called *good examples*, are, and how they should be utilized in learning.

In this context, Freivalds, Kinber, and Wiehagen [41] have introduced two models of learning from good examples, one in the context of finite learning, one in the context of learning in the limit.

Definition 24 (Freivalds *et. al.* [41]). *Let $\mathcal{U} \subseteq \mathcal{R}$ and let $\psi \in \mathcal{P}^2$. The class $\mathcal{U}$ is said to be*

- finitely learnable from good examples with respect to ψ *if there is a numbering $ex \in \mathcal{P}^2$, a strategy $S \in \mathcal{P}$, and a function $z \in \mathcal{P}$ such that $\mathcal{U} \subseteq \mathcal{P}_\psi$ and, for any $i \in \mathbb{N}$ with $\psi_i \in \mathcal{U}$,*
  (1) *ex_i is a finite subfunction of ψ_i and $z(i) = |\{x \mid \psi_i(x)\downarrow\}|$,*
  (2) *$\psi_{S(ex \cup \varepsilon)} = \psi_i$ for any finite subfunction ε of ψ_i.*
- learnable in the limit from good examples with respect to ψ *if there is a numbering $ex \in \mathcal{P}^2$, a strategy $S \in \mathcal{P}^2$, and a function $z \in \mathcal{P}$ such that $\mathcal{U} \subseteq \mathcal{P}_\psi$ and, for any $i \in \mathbb{N}$ with $\psi_i \in \mathcal{U}$,*
  (1) *ex_i is a finite subfunction of ψ_i and $z(i) = |\{x \mid \psi_i(x)\downarrow\}|$,*
  (2) *for any finite subfunction ε of ψ_i, there is some j with $\psi_j = \psi_i$ and $S(ex \cup \varepsilon, n) = j$ for all but finitely many $n \in \mathbb{N}$.*

The resulting learning types are defined in the usual way and denoted by $\mathcal{GEX}$-$\mathcal{FIN}$ and $\mathcal{GEX}$-$\mathcal{LIM}$, respectively.

Interestingly, $\mathcal{GEX}$-$\mathcal{FIN}$ coincides with the standard inference type of behaviorally correct learning, i.e., the classes learnable finitely from good examples can be characterized as exactly those which are learnable behaviorally correctly.

Theorem 56 (Freivalds *et. al.* [41]). $\mathcal{GEX}\text{-}\mathcal{FIN} = \mathcal{BC}$

So, accessing good examples, learning strategies can, even within finite learning processes, achieve more than ordinary strategies can in the limit. This theorem naturally raises the question whether or not $\mathcal{GEX}$-$\mathcal{LIM}$ is richer than $\mathcal{GEX}$-$\mathcal{FIN}$. The affirmative answer is provided by the following theorem. When learning from good examples in the limit, the whole class of all recursive functions can be identified.

Theorem 57 (Freivalds *et. al.* [41]). $\mathcal{R} \in \mathcal{GEX}\text{-}\mathcal{LIM}$

For the proofs of Theorems 56 and 57 as well as for further results, we refer the reader to [41].

The paradigm of learning from good examples has received further interest in research meanwhile; the reader is referred to Nessel [76] for further results on learning recursive functions. Additionally, this framework has been discussed for learning recursive languages as well; here the reader is directed to the survey by Lange *et al.* [69]. It has also attracted interest in other branches of learning theory, see e.g., Ling [72].

9.3. *Complexity of Learning Problems*

So far we have discussed many different learning models derived from Gold's [48] initial one, compared these to one another, and illustrated their strengths and limitations with several examples of learnable or non-learnable classes, respectively. However, we still do not have a deeper insight into what makes some classes harder to learn than others. Characterization theorems provide necessary and sufficient conditions for learnability, but in case two classes are learnable, can we say anything about which of the two is the more challenging learning problem?

To deal with this question, Freivalds *et al.* [39] have introduced a notion of intrinsic complexity of learning. The idea was to define an appropriate notion of reducibility, inspired by results from recursion theory and complexity theory where similar approaches have been successfully applied.

The notion of reducibility defined is based on the usage of recursive operators, i.e., partial recursive operators that are defined on every partial function.

Definition 25 (Rogers [86]). *A partial recursive operator* $\mathfrak{O}\colon \mathfrak{P} \mapsto \mathfrak{P}$ *is called a* recursive operator *if* $\mathtt{dom}(\mathfrak{O}) = \mathfrak{P}$.

We continue with the reduction principle. Given two classes $\mathcal{U}_1$ and $\mathcal{U}_2$ of recursive functions a reduction from $\mathcal{U}_1$ to $\mathcal{U}_2$ involves two recursive operators. The first one translates any function in $\mathcal{U}_1$ into a function in $\mathcal{U}_2$. The second operator converts any successful hypothesis sequence for the obtained function in $\mathcal{U}_2$ into a successful hypothesis sequence for the original function in $\mathcal{U}_1$. Here the term *successful hypothesis sequence* is used to refer to the notion of admissible sequences defined as follows.

Definition 26 (Freivalds *et al.* [39]). *Let* $f \in \mathcal{R}$. *An infinite sequence* σ *is called* $\mathcal{LIM}$-admissible for f *if* σ *converges to a* φ*-program for* f.

This finally allows us to define the desired reducibility relation.

Definition 27 (Freivalds *et al.* [39], Kinber *et al.* [60], Jain *et al.* [54]). *Let* $\mathcal{U}_1, \mathcal{U}_2 \in \mathcal{LIM}$. $\mathcal{U}_1$ *is called* $\mathcal{LIM}$-reducible *to* $\mathcal{U}_2$ *if there exist recursive operators* Θ *and* Ξ *such that each function* $f \in \mathcal{U}_1$ *satisfies the following two conditions:*

(1) $\Theta(f) \in \mathcal{U}_2$,

(2) *if* σ *is a* $\mathcal{LIM}$*-admissible sequence for* $\Theta(f)$*, then* $\Xi(\sigma)$ *is a* $\mathcal{LIM}$*-admissible sequence for* f.

In the sequel we omit the prefix $\mathcal{LIM}$, since we consider the notion of reducibility here only in the context of learning in the limit; however, this definition has also been adapted and analyzed for other inference types.

Now the idea for reductions is as follows: if $\mathcal{U}_1$ is reducible to $\mathcal{U}_2$, then a strategy identifying all functions in $\mathcal{U}_1$ can be computed from any strategy which is successful for $\mathcal{U}_2$. For instance, each class in $\mathcal{LIM}$ is reducible to the class $\mathcal{U}_0$ of functions of finite support (cf. Theorem 58). Thus $\mathcal{U}_0$ is a class in $\mathcal{LIM}$ of highest complexity respecting the notion of reducibility. Such classes are said to be $\mathcal{LIM}$-*complete.*

Definition 28 (Freivalds *et al.* [39], Kinber *et al.* [60], Jain *et al.* [54]). *Let* $\mathcal{U} \subseteq \mathcal{R}$. $\mathcal{U}$ *is* $\mathcal{LIM}$-complete *if* $\mathcal{U} \in \mathcal{LIM}$ *and every class* $\mathcal{U}' \in \mathcal{LIM}$ *is* $\mathcal{LIM}$*-reducible to* $\mathcal{U}$.

Theorem 58 (Freivalds *et. al.* [39]). $\mathcal{U}_0$ is $\mathcal{LIM}$-complete.

Note that $\mathcal{U}_0$ is an r.e. class and every initial segment of any function $f \in \mathcal{U}_0$ is an initial segment of infinitely many other functions $f' \in \mathcal{U}_0$. This property has turned out to be crucial when trying to characterize the hardest problem sets in learning in the limit (with respect to the proposed complexity notion). This is reflected in the following characterization theorem for complete classes.

Theorem 59 (Kinber *et al.* [60], Jain *et al.* [54]). *Let* $\mathcal{U} \in \mathcal{LIM}$. $\mathcal{U}$ *is* $\mathcal{LIM}$*-complete, iff there is some* $\psi \in \mathcal{R}^2$ *such that*

(1) $\mathcal{P}_\psi \subseteq \mathcal{U}$,

(2) *for all* $i, n \in \mathbb{N}$ *there are infinitely many* $j \in \mathbb{N}$ *satisfying* $\psi_i =_n \psi_j$ *and* $\psi_i \neq \psi_j$.

That means that complete classes are characterized by being topologically complicated (in terms of the second condition demanding density with respect to the so-called Baire metric, see Rogers [86] for details) and containing an algorithmically rather "simple" – since uniformly r.e. – set (in terms of the first condition).

9.4. *Uniformity of Learning Problems*

Throughout this survey, we have very often – sometimes implicitly – encountered cases in which different classes of recursive functions can be learned with very similar strategies, or – more specifically – in which different learning problems are solved with a uniform method.

For instance, all classes in $\mathcal{NUM}$ can be learned with Gold's identification by enumeration strategy [48] (cf. the proof of Theorem 2). This strategy is uniform in the sense that it only needs access to the numbering $\psi \in \mathcal{R}^2$ comprising the class of target functions to be learned. Numerous other impressive examples of strategies working uniformly for a huge collection of learning problems have been obtained in the characterization theorems discussed above. Note that in each of these theorems, the sufficiency part of the proof deploys a uniform strategy specific for the inference type that is being characterized.

In contrast to that, most of the learning types studied so far have the property that their corresponding collections of learnable function classes are not closed under union. For instance, the proof of Theorem 19 shows that, although both $\mathcal{U}_{sd}$ and $\mathcal{U}_0$ are learnable in the limit, their union is not even $\mathcal{BC}$-learnable. In particular, both classes can be learned in the limit with two different instances of the same uniform method (different special strategies derived from a kind of *meta-strategy*), but there is no way of designing an instance of that uniform method, which can cope with all functions contained in any of the two classes.

Intuitively, each special strategy for a special class of functions is designed using some prior knowledge about the target class (e.g., the identification by enumeration strategy for a class in $\mathcal{NUM}$ knows a numbering $\psi \in \mathcal{R}^2$ comprising the target class) which can be seen as a restriction of the space of possible hypotheses and thus a restriction of the search space. Now, if the union of two such learnable classes is no longer learnable, this means that there is no means of successfully exploiting the prior knowledge that a function is contained in one of these classes.

Hence there may be ways of describing prior knowledge about certain learnable classes such that this prior knowledge can be exploited by a meta-strategy in order to instantiate strategies tailored for the corresponding target classes. The circumstances under which such meta-strategies exist have been the focus of a branch of research concerning so-called *uniform learning*.

Any formalization of this approach requires a notion of how to describe target classes of recursive functions (i.e., means of describing prior knowledge about a target class) as well as a notion of the desired learning behavior of meta-strategies.

A straightforward scheme for describing classes of recursive functions is the following: Consider a fixed three-place Gödel numbering τ. For any $d \in \mathbb{N}$, the numbering τ^d is just the two-place function resulting from τ, if the first input is fixed by d[1]. Thus

[1] Note that throughout this subsection, for any $d, i \in \mathbb{N}$, we actually denote the function

d corresponds to the set $\mathcal{P}_{\tau^d} = \{\tau_i^d \mid i \in \mathbb{N}\}$ of partial recursive functions enumerated by τ^d and may simply serve as a description of the class $\mathcal{R}_{\tau^d} = \{\tau_i^d \mid i \in \mathbb{N}\} \cap \mathcal{R}$ of recursive functions, which is also called the *recursive core* of the numbering τ^d. In particular, each set $D = \{d_0, d_1, d_2, \ldots\} \subseteq \mathbb{N}$ can be regarded as a set of descriptions and thus as a collection of the "learning problems" $\mathcal{R}_{\tau^{d_0}}, \mathcal{R}_{\tau^{d_1}}, \mathcal{R}_{\tau^{d_2}}, \ldots$ In this context, such a set D is simply called a *description set.*

Now a *meta-strategy* is a strategy expecting two inputs: first, a parameter $d \in \mathbb{N}$ interpreted as a description of some recursive core, and second, a coding f^n of an initial segment of some recursive function f. If S is a meta-strategy and d any description, then S_d denotes the strategy resulting from S, when the first input is fixed by d. Given a learning type LT as studied above, a meta-strategy S is a successful uniform strategy for D, in case S_d learns $\mathcal{R}_{\tau^d}$ for all $d \in D$ according to the constraints of LT. More formally:

Definition 29 (Zilles [110], Jantke [58]). *Let* LT *be a learning type and* $D \subseteq \mathbb{N}$ *a description set. Let* φ *be a Gödel numbering.* D *is* uniformly LT-learnable *(with respect to* φ*) if there is a meta-strategy* S*, such that, for any* $d \in D$*, the strategy* S_d *is an* LT*-learner for the class* $\mathcal{R}_{\tau^d}$ *with respect to* φ*.* $\mathcal{UNI}$-LT *denotes the class of all uniformly* LT*-learnable description sets.*[2]

Lemma 2 justifies the choice of a fixed Gödel numbering φ as a hypothesis space beforehand; the reader may easily verify that this definition of learnability is independent of the choice of φ. However, this is not the only suggestive notion of hypothesis spaces in uniform learning. Note that each numbering τ^d enumerates at least all functions in $\mathcal{R}_{\tau^d}$, so a meta-strategy might also be designed for using τ^d as a hypothesis space when learning $\mathcal{R}_{\tau^d}$. This results in a special case of Definition 29, because τ^d-programs can be uniformly translated into programs in a fixed Gödel numbering φ. We refer to this model as *description-uniform learning.*

Definition 30 (Zilles [110], Jantke [58]). *Let* LT *be a learning type and* $D \subseteq \mathbb{N}$ *a description set.* D *is* description-uniformly LT-learnable*, iff there is a meta-strategy* S*, such that, for any* $d \in D$*, the strategy* S_d *is an* LT*-learner for the class* $\mathcal{R}_{\tau^d}$ *with respect to* τ^d*.* $\mathcal{UNI}_{des}$-LT *denotes the class of all description sets which are description-uniformly* LT*-learnable.*

Uniform learning has been studied for numerous learning types, especially in comparison of different learning types to one another, cf. Zilles [112, 111]; however, for the sake of brevity, we restrict the following survey to the inference types $\mathcal{LIM}$ and $\mathcal{BC}$. Note that most of the results hold analogously for many other learning types. The following theorem states that description-uniform learning is a proper restriction of uniform learning.

$\lambda x.\tau(d, i, x)$ by τ_i^d. Here the superscript d does *not* mean we consider a coding of an initial segment of any function. However, the intended meaning will always be clear from context.

[2] Note that, by intuition, it seems adequate to refer to uniformly learnable *collections of recursive cores* represented by description sets, rather than to uniformly learnable *description sets* themselves. Yet, for convenience, the latter notion is preferred.

Theorem 60 (Zilles [112]).

$\mathcal{UNI}_{des}$-$\mathcal{LIM} \subset \mathcal{UNI}$-$\mathcal{LIM}$ *and* $\mathcal{UNI}_{des}$-$\mathcal{BC} \subset \mathcal{UNI}$-$\mathcal{BC}$*.*

But what are the general limitations of uniform learning? It turns out that there are "maximally powerful" meta-strategies: with a suitable choice of descriptions *all* non-uniformly learnable classes of functions can be learned uniformly.

Theorem 61 (Zilles [112], Jantke [58]). *Let* $\mathsf{LT} \in \{\mathcal{LIM}, \mathcal{BC}\}$*. Then there exists a description set* $\mathsf{D} \subseteq \mathbb{N}$*, such that*

(1) *for all* $\mathcal{U} \in \mathsf{LT}$ *there is some* $\mathsf{d} \in \mathsf{D}$ *satisfying* $\mathcal{U} \subseteq \mathcal{R}_{\tau^{\mathsf{d}}}$*;*

(2) $\mathsf{D} \in \mathcal{UNI}$-$\mathsf{LT}$*.*

In contrast to this, depending on the choice of the descriptions, even sets describing the most simple recursive cores can be non-learnable. This concerns descriptions of recursive cores each consisting of just one function or even, in the description-uniform model, descriptions for just a single recursive core consisting of just one function. This is a significant difference compared to non-uniform learning, because there arbitrary finite classes are trivially learnable.

Theorem 62 (Zilles [112]).

(1) *Let* $\mathsf{D} = \{\mathsf{d} \in \mathbb{N} \mid |\mathcal{R}_{\tau^{\mathsf{d}}}| = 1\}$ *be the set of all descriptions representing singleton recursive cores. Then* $\mathsf{D} \notin \mathcal{UNI}$-$\mathcal{BC}$*.*

(2) *Fix any recursive function* r *and let* $\mathsf{D} = \{\mathsf{d} \in \mathbb{N} \mid \mathcal{R}_{\tau^{\mathsf{d}}} = \{\mathsf{r}\}\}$ *be the set of all descriptions representing the recursive core* $\{\mathsf{r}\}$*. Then* $\mathsf{D} \notin \mathcal{UNI}_{des}$-$\mathcal{BC}$*.*

Theorems 61 and 62 show how much the choice of descriptions affects uniform learnability. In a slightly more subtle way this is expressed in the following two theorems, which moreover show that the hierarchy of learning types in non-uniform learning is reflected in the uniform framework.

Theorem 63 (Zilles [112]). $\mathcal{UNI}$-$\mathcal{LIM} \subset \mathcal{UNI}$-$\mathcal{BC}$*. In particular, there is a* $\mathsf{D} \subseteq \mathbb{N}$ *satisfying*

(1) $|\mathcal{R}_{\tau^{\mathsf{d}}}| = 1$ *for all* $\mathsf{d} \in \mathsf{D}$*,*

(2) $\mathsf{D} \in \mathcal{UNI}$-$\mathcal{BC} \setminus \mathcal{UNI}$-$\mathcal{LIM}$*.*

Theorem 64 (Zilles [112]). $\mathcal{UNI}_{des}$-$\mathcal{LIM} \subset \mathcal{UNI}_{des}$-$\mathcal{BC}$*. In particular, for any* $\mathsf{r} \in \mathcal{R}$*, there is a* $\mathsf{D} \subseteq \mathbb{N}$ *satisfying*

(1) $\mathcal{R}_{\tau^{\mathsf{d}}} = \{\mathsf{r}\}$ *for all* $\mathsf{d} \in \mathsf{D}$*,*

(2) $\mathsf{D} \in \mathcal{UNI}_{des}$-$\mathcal{BC} \setminus \mathcal{UNI}_{des}$-$\mathcal{LIM}$*.*

Now many of the results obtained in the non-uniform framework of learning can be lifted to the meta-level of uniform and description-uniform learning. We will pick two aspects for illustration: (i) characterizations of learning types as in Section 8 and (ii) intrinsic complexity as in Subsection 9.3. From now on, however, we shall focus exclusively on the learning type $\mathcal{LIM}$.

Most of the characterizations shown above have their immediate counterpart in the context of uniform learning, such as the following characterizations derived from Theorem 49. However, it should be noted that characterizations for description-uniform learning always require some additional "embedding" property (like Property (3) in Theorem 66 below). The proofs can be easily lifted from the non-uniform case.

Theorem 65 (Zilles [111]). *Let* $D \subseteq \mathbb{N}$. $D \in \mathcal{UNI}$-$\mathcal{LIM}$, *if and only if there is a three-place partial recursive numbering* χ *and a recursive function* h, *such that for all* $d \in D$ *the following conditions are fulfilled.*

(1) $\mathcal{R}_{\tau^d} \subseteq \mathcal{P}_{\chi^d}$,

(2) $\chi^d_i \neq_{h(d,i,j)} \chi^d_j$ *for all* $i, j \in \mathbb{N}$ *with* $i \neq j$.

Theorem 66 (Zilles [111]). *Let* $D \subseteq \mathbb{N}$. $D \in \mathcal{UNI}_{des}$-$\mathcal{LIM}$, *if and only if there is a three-place partial recursive numbering* χ, *a recursive function* h, *as well as a recursive function* $e \in \mathcal{R}^2$, *such that for all* $d \in D$ *the following conditions are fulfilled.*

(1) $\mathcal{R}_{\tau^d} \subseteq \mathcal{P}_{\chi^d}$,

(2) $\chi^d_i \neq_{h(d,i,j)} \chi^d_j$ *for all* $i, j \in \mathbb{N}$ *with* $i \neq j$,

(3) $\tau^d_{e(d,i)} = \chi^d_i$ *for all* i *such that* $\chi^d_i \in \mathcal{R}$.

In order to lift the intrinsic complexity approach to uniform learning, first appropriate notions of admissible sequences and of reducibility need to be defined – again distinguishing between uniform learning (the $\mathcal{UNI}$-$\mathcal{LIM}$-model) and description-uniform learning (the $\mathcal{UNI}_{des}$-$\mathcal{LIM}$-model). For that purpose, from now on, let φ denote a *fixed* Gödel numbering.

Definition 31 (Zilles [113]). *Let* $d \in \mathbb{N}$ *be any description and let* $f \in \mathcal{R}_{\tau^d}$. *An infinite sequence* σ *of natural numbers is called*

- $\mathcal{UNI}$-$\mathcal{LIM}$-admissible *for* d *and* f *if* σ *converges to a* φ*-program for* f;
- $\mathcal{UNI}_{des}$-$\mathcal{LIM}$-admissible *for* d *and* f *if* σ *converges to a* τ^d*-program for* f.

Since the recursive operators needed now have to take the descriptions $d \in \mathbb{N}$ of the target classes into account, actually a new notion of recursive operators is required.

Definition 32 (Zilles [113]). *Let* Θ *be a total function mapping pairs of functions to pairs of functions.* Θ *is called a* recursive meta-operator *if the following properties are satisfied for all functions* δ, δ' *and* f, f' *and all numbers* $n, y \in \mathbb{N}$:

(1) *if* $\delta \subseteq \delta'$, $f \subseteq f'$, *as well as* $\Theta(\delta, f) = (\gamma, g)$ *and* $\Theta(\delta', f') = (\gamma', g')$, *then* $\gamma \subseteq \gamma'$ *and* $g \subseteq g'$;

(2) *if* $\Theta(\delta, f) = (\gamma, g)$ *and* $\gamma(n) = y$ *(or* $g(n) = y$, *respectively), then there exist initial subfunctions* $\delta_0 \subseteq \delta$ *and* $f_0 \subseteq f$ *such that* $(\gamma_0, g_0) = \Theta(\delta_0, f_0)$ *fulfills* $\gamma_0(n) = y$ *(*$g_0(n) = y$, *respectively);*

(3) *if* δ *and* f *are finite and* $\Theta(\delta, f) = (\gamma, g)$, *then one can effectively (in* (δ, f)*) enumerate* γ *and* g.

This finally enables us to define the following formalization of reducibility in uniform and description-uniform learning.

Definition 33 (Zilles [113]). *Let* $D_1, D_2 \in \mathcal{UNI}\text{-}\mathcal{LIM}$. *We say that* D_1 *is* $\mathcal{UNI}$-$\mathcal{LIM}$-reducible *to* D_2 *if there is a recursive meta-operator* Θ *and a recursive operator* Ξ, *such that the following holds. For any description* $d_1 \in D_1$, *any function* $f_1 \in \mathcal{R}_{d_1}$, *and any* $\delta_1 \in \mathbb{N}^*$ *there is a* $\delta_2 \in \mathbb{N}^*$ *and a function* f_2 *satisfying:*

(1) $\Theta(\delta_1 d_1^\infty, f_1) = (\delta_2, f_2)$,

(2) δ_2 *converges to a description* $d_2 \in D_2$ *such that* $f_2 \in \mathcal{R}_{d_2}$,

(3) *if* σ *is a* $\mathcal{UNI}$-$\mathcal{LIM}$*-admissible sequence for* d_2 *and* f_2, *then* $\Xi(\sigma)$ *is* $\mathcal{UNI}$-$\mathcal{LIM}$*-admissible for* d_1 *and* f_1.

Moreover, if $D_1, D_2 \in \mathcal{UNI}_{des}\text{-}\mathcal{LIM}$, *then the definition of* $\mathcal{UNI}_{des}$-$\mathcal{LIM}$-reducibility *is obtained by replacing* $\mathcal{UNI}$-$\mathcal{LIM}$ *by* $\mathcal{UNI}_{des}$-$\mathcal{LIM}$ *above as well as replacing Condition (3) by Condition (3'): if* σ *is a* $\mathcal{UNI}_{des}$-$\mathcal{LIM}$*-admissible sequence for* d_2 *and* f_2, *then* $\Xi(d_2\sigma)$ *is* $\mathcal{UNI}_{des}$-$\mathcal{LIM}$*-admissible for* d_1 *and* f_1.

As in the non-uniform framework, these reducibility notions immediately yield completeness notions, obtained in the straightforward way.

Definition 34 (Zilles [113]). *Let* $D \subseteq \mathbb{N}$.

(1) D is $\mathcal{UNI}$-$\mathcal{LIM}$-complete *if* $D \in \mathcal{UNI}\text{-}\mathcal{LIM}$ *and every set* $D' \in \mathcal{UNI}\text{-}\mathcal{LIM}$ *is* $\mathcal{UNI}$-$\mathcal{LIM}$*-reducible to* D.

(2) D is $\mathcal{UNI}_{des}$-$\mathcal{LIM}$-complete *if* $D \in \mathcal{UNI}_{des}\text{-}\mathcal{LIM}$ *and every set* $D' \in \mathcal{UNI}_{des}\text{-}\mathcal{LIM}$ *is* $\mathcal{UNI}_{des}$-$\mathcal{LIM}$*-reducible to* D.

For illustration consider the following two examples taken from [113]:

Assume a recursive function g, given any $i, x \in \mathbb{N}$, fulfills $\tau_0^{g(i)} = \varphi_i$ and $\tau_x^{g(i)}\uparrow$, if $x > 0$. Then the description set $\{g(i) \mid i \in \mathbb{N}\}$ is $\mathcal{UNI}$-$\mathcal{LIM}$-complete.

Let $r, h \in \mathcal{R}$. Assume $\tau_i^{h(i)} = r$ for all i, as well as $\varphi_x^{h(i)}\uparrow$ for any $i, x \in \mathbb{N}$ with $x \neq i$. Then the description set $\{h(i) \mid i \in \mathbb{N}\}$ is $\mathcal{UNI}_{des}$-$\mathcal{LIM}$-complete, but not $\mathcal{UNI}$-$\mathcal{LIM}$-complete.

Finally, as in the non-uniform case, we obtain characterizations of $\mathcal{UNI}$-$\mathcal{LIM}$-complete description sets as well as of $\mathcal{UNI}_{des}$-$\mathcal{LIM}$-complete description sets.

Theorem 67 (Zilles [113]). *Let* $D \in \mathcal{UNI}$-$\mathcal{LIM}$. D *is* $\mathcal{UNI}$-$\mathcal{LIM}$*-complete, if and only if there is a* $\psi \in \mathcal{R}^2$ *and a limiting r.e. family* $(d_i)_{i\in\mathbb{N}}$ *of descriptions in* D*, such that the following conditions are fulfilled:*

(1) $\psi_i \in \mathcal{R}_{d_i}$ *for all* $i \in \mathbb{N}$*;*

(2) *every function in* $\mathcal{P}_\psi$ *is an accumulation point of* $\mathcal{P}_\psi$*.*

Theorem 68 (Zilles [113]). *Let* $D \in \mathcal{UNI}_{des}$-$\mathcal{LIM}$. D *is* $\mathcal{UNI}_{des}$-$\mathcal{LIM}$*-complete, if and only if there is a* $\psi \in \mathcal{R}^2$ *and a limiting r.e. family* $(d_i)_{i\in\mathbb{N}}$ *of descriptions in* D*, such that the following conditions are fulfilled:*

(1) $\psi_i \in \mathcal{R}_{d_i}$ *for all* $i \in \mathbb{N}$*;*

(2) *for each* $i, n \in \mathbb{N}$ *there are infinitely many* $j \in \mathbb{N}$ *satisfying* $\psi_i =_n \psi_j$ *and* $(d_i, \psi_i) \neq (d_j, \psi_j)$.

How closely the results on intrinsic complexity of uniform learning are related to the results in the non-uniform framework, as presented in the previous section, is shown in an alternative formulation of Theorem 67, which holds analogously if Property (2) is replaced by the following property: $(2')$ $\mathcal{P}_\psi$ is $\mathcal{LIM}$-complete.

10. Summary and Conclusions

Inductive inference of recursive functions has attracted much attention during the past four decades. As we have seen, the theory has been developed to a large extent and many interesting results have been obtained. These results in turn deepen our principal understanding of inference processes and have many implications for the philosophy of science, for cognition, and of course for learning in general (cf. [79, 55]).

For example, the prediction model (see Definition 4) originating in inductive inference of recursive functions has been adapted in several branches of learning theory. A prominent example is Littlestone's [73] on-line prediction model which is also known as the mistake-bound learning model and which in turn has nice relations to PAC learning (cf. Haussler *et al.* [51]). In this context we mention here again the Algorithm $\mathcal{FP}$ from the proof of Theorem 6.

Furthermore, we have investigated consistent learning versus inconsistent learning, observing a general inconsistency phenomenon: In spite of the remarkable power of consistent learning it turns out that this power is not universal. There are learning problems which can exclusively be solved by inconsistent strategies, i.e., by strategies that do temporarily incorrectly reflect the behavior of the unknown object on data for which the correct behavior of the object is already known. Moreover, the necessity of inconsistent strategies, working in a somewhat unintuitive way, has been traced back to the undecidability of consistency.

If consistent learning is possible, then the corresponding consistency problem must be decidable with respect to some suitably chosen numbering. Further results show that the inconsistency phenomenon is also valid in more realistic situations. Wiehagen and Zeugmann [104] considered a domain where consistency is decidable and the learning strategies have to work in polynomial time, observing that certain learning problems can be solved efficiently, but not efficiently by any consistent strategy (unless $\mathcal{P} = \mathcal{NP}$). The reason is quite analogous to that in the setting of learning recursive functions. Now the $\mathcal{NP}$-hardness of problems can prevent learning strategies from producing consistent hypotheses in polynomial time.

The characterizations of learning types in terms of computable numberings also provide deeper insight concerning the way in which learning strategies can actually perform the inference process in a uniform manner. The first and very powerful method in this regard is Gold's [48] "identification by enumeration." Though the methods provided in the sufficiency proofs of the characterizations theorems in terms of computable numberings have their peculiarities, they all have a rather strong resemblance to "identification by enumeration." This has been further investigated by Kurtz *et al.* [66], where the authors concluded with the thesis that enumeration techniques are even *universal* in that each solvable learning problem in inductive inference can be solved by an adequate enumeration technique. This insight is of fundamental epistemological importance.

On the other hand, these methods do not yield efficient practical algorithms. In general the size of the space that must be searched is typically exponential in the length of the description of the first correct hypothesis.

Looking at the characterizations in terms of complexity, we see that the results obtained also have nice implications for the theory of computational complexity. For illustration, let us recall that there is no function $h \in \mathcal{R}^2$ such that $\Phi_i(x) \leqslant h(x, \varphi_i(x))$ for all $i \in \mathbb{N}$ with $\varphi_i \in \mathcal{R}$ and almost all $x \in \mathbb{N}$ (cf., e.g., [88]). Using Theorem 35 we can directly generalize this to the statement that there is no effective Operator $\mathfrak{O}$ such that $\Phi_i(x) \leqslant \mathfrak{O}(\varphi_i, x)$ for all $i \in \mathbb{N}$ with $\varphi_i \in \mathcal{R}$ and almost all $x \in \mathbb{N}$.

Furthermore, Theorems 35 and 36 together solve the problem of characterizing the operator honest functions. Similar results can be obtained by combining further characterizations (cf. [106]).

Additionally, the characterizations in terms of complexity also show that every function to be learned possesses a recursively computable upper bound for its complexity. In turn, the additional knowledge of such an upper bound does not only guarantee the learnability of the considered functions but also the synthesis of a program with a complexity not greater than the given upper bound almost everywhere.

Let us finish this survey with the remark that the field of learning recursive functions is large but the discourse here is brief. Although several topics could not be touched at all, the material selected for this survey hopefully provides a good overview and guides the reader to other, more specialized sources of information.

References

[1] L. M. Adleman and M. Blum. Inductive inference and unsolvability. *The Journal of Symbolic Logic*, 56(3):891–900, 1991.

[2] Y. Akama and T. Zeugmann. Consistency conditions for inductive inference of recursive functions. In *New Frontiers in Artificial Intelligence, JSAI 2006 Conference and Workshops, Tokyo, Japan, June 2006, Revised Selected Papers*, volume 4384 of *Lecture Notes in Artificial Intelligence*, pages 251–264, Berlin, 2007. Springer.

[3] A. Ambainis. Probabilistic inductive inference: a survey. *Theoret. Comput. Sci.*, 264(1):155–167, 2001.

[4] A. Ambainis and R. Freivalds. Transformations that preserve learnability. In *Algorithmic Learning Theory, 7th International Workshop, ALT '96, Sydney, Australia, October 1996, Proceedings*, volume 1160 of *Lecture Notes in Artificial Intelligence*, pages 299–311. Springer, 1996.

[5] D. Angluin and C. H. Smith. A survey of inductive inference: Theory and methods. *ACM Comput. Surv.*, 15(3):237–269, 1983.

[6] D. Angluin and C. H. Smith. Inductive inference. In *Encyclopedia of Artificial Intelligence*, pages 409–418. J. Wiley and Sons, New York, 1987.

[7] K. Apsītis, R. Freivalds, R. Simanovskis, and J. Smotrovs. Closedness properties in ex-identification. *Theoret. Comput. Sci.*, 268(2):367–393, 2001.

[8] K. Apsītis, R. Freivalds, and C. Smith. Choosing a learning team: a topological approach. In *Proc. of Twenty-sixth ACM Symposium on Theory of Computing*, 1994.

[9] J. M. Barzdin. Prognostication of automata and functions. In C. V. Freiman, J. E. Griffith, and J. L. Rosenfeld, editors, *Information Processing 71, Proceedings of IFIP Congress 71, Volume 1 - Foundations and Systems, Ljubljana, Yugoslavia, August 23-28, 1971*, pages 81–84. North-Holland, 1972.

[10] J. M. Barzdin. Inductive inference of automata, functions and programs. In *Proc. of the 20-th International Congress of Mathematicians, Vancouver, Canada*, pages 455–460, 1974. (republished in *Amer. Math. Soc. Transl.* (2) 109, 1977, pages 107- 112).

[11] J. M. Barzdin, editor. Теория Алгоритмов и Программ, volume I. Latvian State University, 1974.

[12] J. M. Barzdin, editor. Теория Алгоритмов и Программ, volume II. Latvian State University, 1975.

[13] J. M. Barzdin, editor. Теория Алгоритмов и Программ, volume III. Latvian State University, 1977.

[14] J. M. Barzdin and R. V. Freivald. On the prediction of general recursive functions. *Soviet Math. Dokl.*, 13:1224–1228, 1972.

[15] Я. М. Барздинь. Сложность и частотное решение некоторых алгоритмически неразрешимых массовых проблем. Докт. диссертация, Новосибирск, 1971.

[16] Я. М. Барздинь. Две теоремы о предельном синтезе функций. In J. M. Barzdin, editor, Теория Алгоритмов и Программ, volume I, pages 82–88. Latvian State University, 1974.

[17] Я. М. Барздинь and Р. В. Фрейвалд. Прогнозирование и предельный синтез еффективно перечислимых классов функций. In J. M. Barzdin, editor, Теория Алгоритмов и Программ, volume I, pages 101–111. Latvian State University, 1974.

[18] Я. М. Барздинь and К. М. Подниекс. К теорию индуктивного вывода. In *Mathematical Foundations of Computer Science: Proceedings of Symposium and Summer School, Strbské Pleso, High Tatras, Czechoslovakia, September 3-8, 1973.*, pages 9–15. Mathematical Institute of the Slovak Academy of Sciences, 1973.

[19] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Inform. Control*, 28(2):125–155, June 1975.

[20] M. Blum. A machine independent theory of the complexity of recursive functions. *Journal of the ACM*, 14(2):322–336, 1967.

[21] R. Board and L. Pitt. On the necessity of Occam algorithms. *Theoret. Comput. Sci.*, 100(1):157–184, 1992.

[22] A. Borodin. Computational complexity and the existence of complexity gaps. *Journal of the ACM*, 19(1):158–174, 1972.

[23] U. Brandt. The position of index sets of identifiable sets in the arithmetical hierarchy. *Inform. Control*, 68(1-3):185–195, 1986.

[24] J. Case, S. Jain, S. Lange, and T. Zeugmann. Incremental concept learning for bounded data mining. *Inform. Comput.*, 152(1):74–110, 1999.

[25] J. Case, S. Jain, M. Ott, A. Sharma, and F. Stephan. Robust learning aided by context. *J. Comput. Syst. Sci.*, 60(2):234–257, 2000.

[26] J. Case, S. Jain, F. Stephan, and R. Wiehagen. Robust learning – rich and poor. *J. Comput. Syst. Sci.*, 69(2):123–165, 2004.

[27] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoret. Comput. Sci.*, 25(2):193–220, 1983.

[28] R. L. Constable. The operator gap. *Journal of the ACM*, 19(1):175–183, 1972.

[29] R. P. Daley. On the error correcting power of pluralism in BC-type inductive inference. *Theoret. Comput. Sci.*, 24(1):95–104, 1983.

[30] R. P. Daley. Towards the development of an analysis of learning algorithms. In *Analogical and Inductive Inference, International Workshop AII '86. Wendisch-Rietz, GDR, October 1986, Proceedings*, volume 265 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 1986.

[31] J. Feldman. Some decidability results on grammatical inference and complexity. *Inform. Control*, 20(3):244–262, 1972.

[32] R. V. Freivald and R. Wiehagen. Inductive inference with additional information. *Elektronische Informationsverarbeitung und Kybernetik*, 15(4):179–185, 1979.

[33] Р. В. Фрейвалд. Предельно вычислимые функции и функционалы. In J. M. Barzdin, editor, Теория Алгоритмов и Программ, volume I, pages 6–19. Latvian State University, 1974.

[34] Р. В. Фрейвалд and Я. М. Барздинь. Соотношения между прогнозируемостью и предельной синтезируемостью. In J. M. Barzdin, editor, Теория Алгоритмов и Программ, volume II, pages 26 – 34. Latvian State University, 1975.

[35] R. Freivalds. An answer to an open problem. *Bulletin of the EATCS*, 23:31–32, 1984.

[36] R. Freivalds. Inductive inference of minimal programs. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 3–20, San Mateo, CA, 1990. Morgan Kaufmann.

[37] R. Freivalds, J. Bārzdiņš, and K. Podnieks. Inductive inference of recursive functions: Complexity bounds. In *Baltic Computer Science*, volume 502 of *Lecture Notes in Computer Science*, pages 111–155. Springer-Verlag, Berlin, 1991.

[38] R. Freivalds, O. Botuscharov, and R. Wiehagen. Identifying nearly minimal Gödel numbers from additional information. *Annals of Mathematics and Artificial Intelligence*, 23(1/2):199–209, 1998.

[39] R. Freivalds, E. Kinber, and C. H. Smith. On the intrinsic complexity of learning. *Inform. Comput.*, 123(1):64–71, 1995.

[40] R. Freivalds, E. B. Kinber, and R. Wiehagen. Probabilistic versus deterministic inductive inference in nonstandard numberings. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 34:531–539, 1988.

[41] R. Freivalds, E. B. Kinber, and R. Wiehagen. On the power of inductive inference from good examples. *Theoret. Comput. Sci.*, 110(1):131–144, 1993.

[42] R. Freivalds, C. H. Smith, and M. Velauthapillai. Trade-off among parameters affecting inductive inference. *Inform. Comput.*, 82:323–349, 1989.

[43] R. V. Freivalds. Finite identification of general recursive functions by probabilistic strategies. In *Fundamentals of Computations Theory, FCT' 79, Proc. of the Conference on Algebraic, Arithmetic, and Categorial Methods in Computation Theory*, pages 138–145, Berlin, 1979. Akademie-Verlag.

[44] M. A. Fulk. Robust separations in inductive inference. In *Proc. of the 31st Annual Symposium on Foundations of Computer Science*, pages 405–410. IEEE Computer Society Press, Los Alamitos, CA, 1990.

[45] W. Gasarch and C. H. Smith. A survey of inductive inference with an emphasis on queries. In *Complexity, Logic, and Recursion Theory.* Marcel Dekker, Inc., 1997.

[46] W. I. Gasarch, R. K. Sitaraman, C. H. Smith, and M. Velauthapillai. Learning programs with an easy to calculate set of errors. *Fundamenta Informaticae*, pages 355–370, 1992.

[47] E. M. Gold. Limiting recursion. *Journal of Symbolic Logic*, 30:28–48, 1965.

[48] E. M. Gold. Language identification in the limit. *Inform. Control*, 10(5):447–474, 1967.

[49] J. Grabowski. Starke Erkennung. In R. Linder and H. Thiele, editors, *Strukturerkennung diskreter kybernetischer Systeme*, volume 82, pages 168–184. Seminarberichte der Sektion Mathematik der Humboldt-Universität zu Berlin, 1986.

[50] G. Grieser. Reflective inductive inference of recursive functions. *Theoret. Comput. Sci.*, xxx(xxx):xx–xx, 2007.

[51] D. Haussler, M. Kearns, N. Littlestone, and M. K. Warmuth. Equivalence of models for polynomial learnability. *Inform. Comput.*, 95(2):129–161, 1991.

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

[53] S. Jain. Robust behaviorally correct learning. *Inform. Comput.*, 153(2):238–248, 1999.

[54] S. Jain, E. Kinber, C. Papazian, C. Smith, and R. Wiehagen. On the intrinsic complexity of learning recursive functions. *Inform. Comput.*, 184(1):45–70, 2003.

[55] S. Jain, D. Osherson, J. S. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory, second edition.* MIT Press, Cambridge, Massachusetts, 1999.

[56] S. Jain and A. Sharma. Learning with the knowledge of an upper bound on program size. *Inform. Comput.*, 102(1):118–166, 1993.

[57] S. Jain, C. Smith, and R. Wiehagen. Robust learning is rich. *J. Comput. Syst. Sci.*, 62(1):178–212, 2001.

[58] K. P. Jantke. Natural properties of strategies identifying recursive functions. *Elektronische Informationsverarbeitung und Kybernetik*, 15(10):487–496, 1979.

[59] K. P. Jantke and H.-R. Beick. Combining postulates of naturalness in inductive inference. *Elektronische Informationsverarbeitung und Kybernetik*, 17(8/9):465–484, 1981.

[60] E. Kinber, C. Papazian, C. Smith, and R. Wiehagen. On the intrinsic complexity of learning recursive functions. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory, July 6th-9th, 1999, Santa Cruz, California*, pages 257–266, New York, NY, 1999. ACM Press.

[61] E. Kinber and T. Zeugmann. One-sided error probabilistic inductive inference and reliable frequency identification. *Inform. Comput.*, 92(2):253–284, 1991.

[62] E. B. Kinber and T. Zeugmann. Inductive inference of almost everywhere correct programs by reliably working strategies. *Elektronische Informationsverarbeitung und Kybernetik*, 21(3):91–100, 1985.

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

[64] S. A. Kurtz and C. H. Smith. On the role of search for learning. In *Proceedings of the Second Annual Workshop on Computational Learning Theory*, pages 303–311, San Mateo, CA, 1989. Morgan Kaufmann.

[65] S. A. Kurtz and C. H. Smith. A refutation of Barzdins' conjecture. In *Analogical and Inductive Inference, International Workshop AII '89, Reinhardsbrunn Castle, GDR, October 1989, Proceedings*, volume 397 of *Lecture Notes in Artificial Intelligence*, pages 171–176. Springer-Verlag, 1989.

[66] S. A. Kurtz, C. H. Smith, and R. Wiehagen. On the role of search for learning from examples. *J. of Experimental and Theoret. Artif. Intell.*, 13(1):25–43, 2001.

[67] L. H. Landweber and E. L. Robertson. Recursive properties of abstract complexity classes. *Journal of the ACM*, 19(2):296–308, 1972.

[68] S. Lange and T. Zeugmann. Incremental learning from positive data. *J. of Comput. Syst. Sci.*, 53(1):88–103, 1996.

[69] S. Lange, T. Zeugmann, and S. Zilles. Learning indexed families of recursive languages from positive data: a survey. *Theoret. Comput. Sci.*, xxx(x):xx–xx, 2007.

[70] R. Linder and H. Thiele, editors. *Strukturerkennung diskreter kybernetischer Systeme*, volume 82. Seminarberichte der Sektion Mathematik der Humboldt-Universität zu Berlin, 1986.

[71] R. Lindner. *Algorithmische Erkennung.* Dissertation B, Friedrich-Schiller-Universität, Jena, 1972.

[72] X. C. Ling. Inductive learning from good examples. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia, August 24-30, 1991*, pages 751–756. Morgan Kaufmann, 1991.

[73] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[74] E. M. McCreight and A. R. Meyer. Classes of computable functions defined by bounds on computation: Preliminary report. In *Proceedings of the first annual ACM symposium on Theory of computing, Marina del Rey, California, United States*, pages 79–88. ACM Press, 1969.

[75] E. Minicozzi. Some natural properties of strong identification in inductive inference. *Theoret. Comput. Sci.*, 2(3):345–360, 1976.

[76] J. Nessel. Learnability of enumerable classes of recursive functions from "typical" examples. In *Algorithmic Learning Theory, 10th International Conference, ALT '99, Tokyo, Japan, December 1999, Proceedings*, volume 1720 of *Lecture Notes in Artificial Intelligence*, pages 264–275. Springer, 1999.

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

[78] W. of Ockham. *Quodlibeta Septem* (in translation). Circa 1320.

[79] D. N. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Massachusetts, 1986.

[80] M. Ott and F. Stephan. Avoiding coding tricks by hyperrobust learning. In *Computational Learning Theory, 4th European Conference, EuroCOLT '99, Nordkirchen, Germany, March 29-31, 1999, Proceedings*, volume 1572 of *Lecture Notes in Artificial Intelligence*, pages 183–197. Springer, 1999.

[81] L. Pitt. Probabilistic inductive inference. *J. ACM*, 36(2):383–433, 1989.

[82] К. М. Подниекс. Сравнение различных типов синтеза и прогнозирования функций I. In J. M. Barzdin, editor, Теория Алгоритмов и Программ, volume I, pages 68–81. Latvian State University, 1974.

[83] К. М. Подниекс. Сравнение различных типов синтеза и прогнозирования функций II. In J. M. Barzdin, editor, Теория Алгоритмов и Программ, volume II, pages 35–44. Latvian State University, 1975.

[84] K. Popper. *The Logic of Scientific Discovery. (2nd ed.).* Harper Torch Books, New York, 1968.

[85] H. Putnam. Probability and confirmation. In *Mathematics, Matter and Method.* Cambridge University Press, 1975.

[86] H. J. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967. Reprinted, MIT Press 1987.

[87] C. H. Smith. The power of pluralism for automatic program synthesis. *J. ACM*, 29(4):1144–1165, 1982.

[88] C. H. Smith. *A Recursive Introduction to the Theory of Computation.* Springer-Verlag, New York, Berlin, Heidelberg, 1994.

[89] C. H. Smith. Three decades of team learning. In *Algorithmic Learning Theory, 4th International Workshop on Analogical and Inductive Inference, AII '94, 5th International Workshop on Algorithmic Learning Theory, ALT '94, Reinhardsbrunn Castle, Germany, October 1994, Proceedings*, volume 872 of *Lecture Notes in Artificial Intelligence*, pages 211–228. Springer-Verlag, 1994.

[90] C. H. Smith and M. Velauthapillai. On the inference of programs approximately computing the desired function. In *Analogical and Inductive Inference, International Workshop AII '86. Wendisch-Rietz, GDR, October 1986, Proceedings*, volume 265 of *Lecture Notes in Computer Science*, pages 164–176. Springer-Verlag, 1986.

[91] R. M. Smullyan. *Theory of formal systems*, volume 47 of *Annals of Mathathematical Studies.* Princeton University Press, Princeton, New Jersey, USA, 1961.

[92] R. J. Solomonoff. A formal theory of inductive inference. part i. *Inform. Control*, 7(1):1–22, 1964.

[93] R. J. Solomonoff. A formal theory of inductive inference. part ii. *Inform. Control*, 7(2):224–254, 1964.

[94] F. Stephan and T. Zeugmann. Learning classes of approximations to non-recursive functions. *Theoret. Comput. Sci.*, 288(2):309–341, 2002.

[95] H. Thiele. Lernverfahren zur Erkennung formaler Sprachen. In *Lernende Systeme*, volume 3 of *Kybernetik-Forschung*, pages 11–93. Deutscher Verlag der Wissenschaften, Berlin, 1973.

[96] B. A. Trakhtenbrot and Y. M. Barzdin. *Finite Automata, Behavior and Synthesis.* North Holland, Amsterdam, 1973.

[97] R. Wiehagen. Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Elektronische Informationsverarbeitung und Kybernetik*, 12(1/2):93–99, 1976.

[98] R. Wiehagen. Characterization problems in the theory of inductive inference. In *Automata, Languages and Programming Fifth Colloquium, Udine, Italy, July 17-21, 1978*, volume 62 of *Lecture Notes in Computer Science*, pages 494 – 508, Berlin, 1978. Springer-Verlag.

[99] R. Wiehagen. *Zur Theorie der Algorithmischen Erkennung.* Dissertation B, Humboldt-Universität zu Berlin, 1978.

[100] R. Wiehagen. A thesis in inductive inference. In *Nonmonotonic and Inductive Logic, 1st International Workshop, Karlsruhe, Germany, December 1990, Proceedings*, volume 543 of *Lecture Notes in Artificial Intelligence*, pages 184–207, Berlin, 1990. Springer-Verlag.

[101] R. Wiehagen, R. Freivalds, and E. Kinber. On the power of probabilistic strategies in inductive inference. *Theoret. Comput. Sci.*, 28(1-2):111–133, 1984.

[102] R. Wiehagen and W. Liepe. Charakteristische Eigenschaften von erkennbaren Klassen rekursiver Funktionen. *Elektronische Informationsverarbeitung und Kybernetik*, 12(8/9):421–438, 1976.

[103] R. Wiehagen and T. Zeugmann. Ignoring data may be the only way to learn efficiently. *J. of Experimental and Theoret. Artif. Intell.*, 6(1):131–144, 1994.

[104] R. Wiehagen and T. Zeugmann. Learning and consistency. In *Algorithmic Learning for Knowledge-Based Systems, GOSLER Final Report*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 1–24. Springer, 1995.

[105] P. Young. Easy constructions in complexity theory: Gap and speed-up theorems. *Proceedings of the American Mathematical Society*, 37(2):555–563, 1973.

[106] T. Zeugmann. A-posteriori characterizations in inductive inference of recursive functions. *Elektronische Informationsverarbeitung und Kybernetik*, 19(10/11):559–594, 1983.

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

[108] T. Zeugmann. On Barzdin's conjecture. In *Analogical and Inductive Inference, International Workshop AII '86. Wendisch-Rietz, GDR, October 1986, Proceedings*, volume 265 of *Lecture Notes in Computer Science*, pages 220–227. Springer-Verlag, 1986.

[109] T. Zeugmann. Inductive inference of optimal programs: A survey and open problems. In *Nonmonotonic and Inductive Logic, 1st International Workshop, Karlsruhe, Germany, December 1990, Proceedings*, volume 543 of *Lecture Notes in Artificial Intelligence*, pages 208–222. Springer-Verlag, 1990.

[110] S. Zilles. On the synthesis of strategies identifying recursive functions. In *Computational Learning Theory: 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 16-19, 2001, Proceedings*, volume 2111 of *Lecture Notes in Artificial Intelligence*, pages 160–176. Springer-Verlag, 2001.

[111] S. Zilles. *Uniform Learning of Recursive Functions.* DISKI 278, Akademische Verlagsgesellschaft Aka GmbH, 2003.

[112] S. Zilles. Separation of uniform learning classes. *Theoret. Comput. Sci.*, 313:229–265, 2004.

[113] S. Zilles. An approach to intrinsic complexity of uniform learning. *Theoret. Comput. Sci.*, 364(1):42–61, 2006.