𝕋ℂ𝕊-𝕋ℝ-𝔸-**11-49**

𝕋ℂ𝕊 Technical Report

On the Amount of Nonconstructivity in the Inductive Inference of Recursive Functions

by

Rūsiņš Freivalds and Thomas Zeugmann

Division of Computer Science

Report Series A

December 17, 2010

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

On the Amount of Nonconstructivity in the Inductive Inference of Recursive Functions

Rūsiņš Freivalds*
Institute of Mathematics and Computer Science
University of Latvia
Raiņa Bulvāris 29
Riga, LV-1459, Latvia
`Rusins.Freivalds@mii.lu.lv`

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

Abstract

Nonconstructive proofs are a powerful mechanism in mathematics. Furthermore, nonconstructive computations by various types of machines and automata have been considered by e.g., Karp and Lipton [15] and Freivalds [10]. They allow to regard more complicated algorithms from the viewpoint of much more primitive computational devices. The amount of nonconstructivity is a quantitative characterization of the distance between types of computational devices with respect to solving a specific problem.

In the present paper, the amount of nonconstructivity in learning of recursive functions is studied. Different learning types are compared with respect to the amount of nonconstructivity needed to learn the whole class of general recursive functions. Upper and lower bounds for the amount of nonconstructivity needed are proved.

Keywords: inductive inference, learning, recursive functions, nonconstructivity, advice

1. Introduction

Nonconstructive methods of proof in mathematics have a rather long and dramatic history. The debate was especially passionate when mathematicians tried to overcome the crisis concerning the foundations of mathematics.

*This research was performed while this author was visiting the Division of Computer Science at Hokkaido University. The research of the first author was supported by Grant No. 09.1570 from the Latvian Council of Science and by Project 2009/0216/1DP/1.1.2.1.2/09/IPIA/VIA/004 from the European Social Fund.

†Supported by MEXT Grant-in-Aid for Scientific Research on Priority Areas under Grant No. 21013001.

However, the situation changed slightly in the forties of the last century, when nonconstructive methods found their way even to discrete mathematics. In particular, Paul Erdős used nonconstructive proofs masterly, beginning with the paper [6].

Another influential paper in this regard was Bārzdiņš [2], who introduced the notion of advice in the setting of Kolmogorov complexity of recursively enumerable sets. Karp and Lipton [15] introduced the notion of a Turing machine that takes advice to understand under what circumstances nonuniform upper bounds can be used to obtain uniform upper bounds.

A further step was taken by Freivalds [10], who introduced a qualitative approach to measure the amount of nonconstructivity (or advice) in a proof. Analyzing three examples of nonconstructive proofs led him to a notion of nonconstructive computation which can be easily used for many types of automata and machines and which essentially coincides with Karp and Lipton's [15] notion when applied to Turing machines.

As outlined by Freivalds [10], there are several results in the theory of inductive inference of recursive functions which suggest that the notion of nonconstructivity may be worth a deeper study in this setting, too.

In the present paper we prove several upper and lower bounds for the amount of nonconstructivity in learning classes of recursive functions. When learning recursive functions growing initial segments $(f(0), \ldots, f(n))$ are fed to the learning algorithm, henceforth called *strategy*. For each initial segment the strategy has then to compute a hypothesis i_n which is a natural number. These hypotheses are interpreted with respect to a suitably chosen hypothesis space ψ which is a numbering. The interpretation of the hypothesis i_n is that the strategy is conjecturing program i_n in the numbering ψ to compute the target function f. One requires the sequence $(i_n)_{n\in\mathbb{N}}$ of all computed hypotheses to converge to a program correctly computing the target function f. A strategy learns a class of recursive functions provided it can learn every function from it. The model just explained is basically *learning in the limit* as introduced by Gold [12]. Many variations of this model have been studied (cf., e.g., [4, 9, 23], and the references therein).

On the one hand, for many of these variations it was shown that the class of all recursive functions is *not* learnable. Also, several attempts have been undertaken to classify the difficulty of learning the class of all recursive functions. Adleman and Blum [1] showed the degree of unsolvability of the problem to learn all recursive functions to be strictly less than the degree of the halting problem. A further approach was to characterize the difficulty of learning classes of recursive functions by using oracles (cf., e.g., [5, 17]).

Our goal is to introduce a different measure, i.e., the amount of nonconstructivity needed to learn all recursive functions. That is, the strategy receives as a second input a bitstring of finite length which we call *help-word*. If the help-word is correct, the strategy learns in the desired sense. Since there are infinitely many functions to learn, a parameterization is necessary, i.e., we allow for every n a possibly different help-word and we require the strategy to learn every recursive function contained in $\{\psi_0, \ldots, \psi_n\}$ with respect to the numbering ψ (cf. Definition 4). The difficulty of the learning problem is then measured by the length of the help-words needed, i.e., in terms of the growth rate of function d bounding this length.

As in previous approaches, the help-word does *not* just provide an answer to the learning problem. There is still much work to be done by the strategy. The usefulness of this approach is nicely reflected by our results which show that the function d may vary from arbitrarily slow growing (for learning in the limit) to $n + 1$ (for minimal identification).

2. Preliminaries

Any unspecified notations follow Rogers [20]. In addition to or in contrast with Rogers [20] we use the following. By $\mathbb{N} = \{0, 1, 2, \ldots\}$ 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 $\mathcal{P}$, $\mathcal{R}$, $\mathcal{P}^2$, and $\mathcal{R}^2$, respectively.

For any function $f \in \mathcal{P}$ 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 *strictly monotonic* provided for all $x, y \in \mathbb{N}$ with $x < y$ we have, if both $f(x)$ and $f(y)$ are defined then $f(x) < f(y)$. By $\mathcal{R}_{mon}$ we denote the set of all strictly monotonic 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. $\psi_i(x)\!\uparrow$). Let $\psi \in \mathcal{P}^2$ be any numbering and let $f \in \mathcal{P}$; then we use $\min_\psi f$ to denote the least number i such that $\psi_i = f$. We refer to $\min_\psi f$ as a ψ-minimal program of f.

A numbering $\varphi \in \mathcal{P}^2$ is called a *Gödel numbering* (cf. Rogers [20]) 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 $i \in \mathbb{N}$. *Göd* denotes the set of all Gödel numberings.

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 and let $\mathcal{NUM}! = \{\mathcal{U} \mid (\exists\psi \in \mathcal{R}^2)\,[\mathcal{U} = \mathcal{P}_\psi]\}$ denote the family of all recursively enumerable classes of recursive functions. Note that the elements of $\mathcal{NUM}!$ are also often referred to as indexed families.

Following [18] 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 [3] 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 $i, x, y \in \mathbb{N}$.

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 $n \in \mathbb{N},\ f \in \mathcal{R}$. 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 $k \geq n$.

Definition 1 (Gold [11, 12]). *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 $\mathcal{U} \in \mathcal{LIM}_\psi(S)$. We set $\mathcal{LIM}_\psi = \{\mathcal{U} \mid \mathcal{U}$ is learnable in the limit with respect to $\psi\}$. Finally, let $\mathcal{LIM} = \bigcup_{\psi\in\mathcal{P}^2} \mathcal{LIM}_\psi$.

Freivalds [8] and Kinber [16] introduced the following modification of Definition 1, where instead of converging to any program for the target function f, the strategy is required to converge to $\min_\psi f$.

Definition 2 (Freivalds [8], Kinber [16]). *Let $\mathcal{U} \subseteq \mathcal{R}$ and let $\psi \in \mathcal{P}^2$. The class $\mathcal{U}$ is said to be* ψ-minimal 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) *the sequence $(S(f^n))_{n\in\mathbb{N}}$ converges to $\min_\psi f$.*

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

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*." This leads to finite identification which has been investigated intensively in the literature (cf., e.g., Jain, Osherson, Royer, and Sharma [13] and Zilles and Zeugmann [23] and the references given therein).

Definition 3 (Gold [12], Trakhtenbrot and Barzdin [21]). *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 the class $\mathcal{U}$ is finitely learnable with respect to ψ by a strategy S then we write $\mathcal{U} \in \mathcal{FIN}_\psi(S)$. Let $\mathcal{FIN}_\psi = \{\mathcal{U} \mid \mathcal{U}$ is finitely learnable with respect to $\psi\}$, and let $\mathcal{FIN} = \bigcup_{\psi\in\mathcal{P}^2} \mathcal{FIN}_\psi$.

Of course, we can also combine ψ-minimal learnability and finite identification. That is, now the strategy has to converge finitely to a ψ-minimal program of the target function. We denote the resulting learning type by $\mathcal{MIN}$-$\mathcal{FIN}_\psi$.

The strategies used for nonconstructive inductive inference take as input not only the encoded graph of a recursive function but also a help-word. The help-words are assumed to be encoded in binary. So, for such strategies we write $S(f^n, w)$ to denote the program output by S, where w is the help-word. Then, for all the inference types defined above, we say that S nonconstructively identifies f with the help-word w provided the sequence $(S(f^n, w))_{n\in\mathbb{N}}$ (finitely) converges to a number j such that $\varphi_j = f$ (for $\mathcal{LIM}$ and $\mathcal{FIN}$) and $j = \min_\psi f$ (for $\mathcal{MIN}$), respectively.

Definition 4. *Let $\psi \in \mathcal{P}^2$, let $\mathcal{U} \subseteq \mathcal{R}$, and let $d \in \mathcal{R}$. A strategy $S \in \mathcal{P}^2$* identifies $\mathcal{U}$ with nonconstructivity $d(n)$ in the limit with respect to ψ, *if for every $n \in \mathbb{N}$ there is a help-word of length at most $d(n)$ such that for every $f \in \mathcal{U} \cap \{\psi_0, \psi_1, \ldots, \psi_n\}$ the sequence $(S(f^n, w))_{n\in\mathbb{N}}$ converges to a program i satisfying $\psi_i = f$.*

Nonconstructive finite and minimal identification are defined in analogue to the above.

Looking at Definition 4 as well as at the definition of nonconstructive finite and minimal identification, it should be noted that the strategy may need to know either an appropriate upper bound for n or even the precise value of n in order to exploit the fact that the target function is from $f \in \mathcal{U} \cap \{\psi_0, \psi_1, \ldots, \psi_n\}$.

In order to simplify notation in several theorems and proofs given below, we make the following convention. Whenever we talk about nonconstructivity $\log n$, we assume that the logarithmic function to the base 2 is replaced by its integer valued counterpart $\lfloor \log n \rfloor + 1$.

3. Results

Already Gold [11] showed that $\mathcal{R} \notin \mathcal{LIM}$. So, we start our investigations by asking for the amount of nonconstructivity needed to identify the set $\mathcal{R}$ of all recursive functions in the limit with respect to any Gödel numbering φ.

Using an idea from Freivald and Wiehagen [7], we prove that the needed amount of nonconstructivity is surprisingly small. To show this result, for every function $f \in \mathcal{R}_{mon}$ we define its *inverse* f_{inv} as follows $f_{inv}(n) = \mu y[f(y) \geq n]$ for all $n \in \mathbb{N}$. Recall that $\mathtt{Val}(f)$ is recursive for all $f \in \mathcal{R}_{mon}$. Thus, for all $f \in \mathcal{R}_{mon}$ we can conclude that $f_{inv}(n) \in \mathcal{R}$.

Theorem 1. *Let $\varphi \in$ Göd be arbitrarily fixed, and let $d \in \mathcal{R}_{mon}$ be any function. Then there is a strategy $S \in \mathcal{P}^2$ such that the class $\mathcal{R}$ can be identified with nonconstructivity $\log d_{inv}(n)$ in the limit with respect to φ.*

Proof. Let $\varphi \in$ *Göd* be arbitrarily fixed. Without loss of generality, we can also assume any complexity function $\Phi \in \mathcal{P}^2$ such that (φ, Φ) is a complexity measure.

The key idea of the proof is that, in order to learn any function from $\mathcal{R}$, it suffices to have an upper bound for $\min_\varphi f$. So, assuming any help-word w of length $\log d_{inv}(n)$, the strategy S uses the length of the help-word w to create a bitstring that contains only 1s and has the same length as the help word. This bitstring is interpreted in the usual way as a natural number k. By construction, we then have $k \geq d_{inv}(n)$. Furthermore, since $d \in \mathcal{R}_{mon}$, we directly obtain that $d(k) \geq d(d_{inv}(n)) \geq n$. Consequently, the strategy S uses k to compute

$$u_* \;=_{df} d(k) \;,$$

and by construction, we have $u_* \geq n$.

Assume any function $f \in \mathcal{R} \cap \{\psi_0, \psi_1, \ldots, \psi_n\}$, and let f^m and w be the input to S. Then, the strategy initializes the index set I_{init} to be $I_{init} = \{0, \ldots, u_*\}$ and checks whether or not $\Phi_i(x) \leq m$ for every $i \in I_{init}$ and $0 \leq x \leq m$. For all i and x that passed this test successfully, the strategy then checks whether or not $\varphi_i(x) = f(x)$. If this is not the case, i is removed from I_{init}. Let I_m be the resulting index set.

Finally, the strategy uses the amalgamation technique (cf. Wiehagen [22], Case and Smith [4]). That is, 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.

So, the output of $S(f^m, w)$ is $\mathrm{amal}(I_m)$.

It remains to show that the sequence $(\mathrm{amal}(I_m))_{m\in\mathbb{N}}$ converges to a φ-program for f. By construction we know that I_{init} contains at least one φ-program for f. Clearly, this program and any other φ-program computing a subfunction of f can never be removed from I_{init}. But if a φ-program j from I_{init} does not compute a subfunction of f, then there must be an x such that $\varphi_j(x)\downarrow \neq f(x)$. So, as soon as $m \geq \max\{x, \Phi_j(x)\}$, the program j is removed from I_{init}. Since I_{init} is finite, there must exists an m_* such that I_{m_*} contains only φ-programs computing f or a subfunction of f. We conclude that $\text{amal}(I_{m_*})$ is a φ-programs for f. Furthermore, $I_\ell = I_{m_*}$ for all $\ell \geq m_*$, and thus the strategy S learns f in the limit. ∎

As we have seen, there is no smallest amount of nonconstructivity needed to learn $\mathcal{R}$ in the limit. On the other hand, the amount of nonconstructivity cannot be zero, since then we would have $\mathcal{R} \in \mathcal{LIM}$. But one can define a total function $t \in \mathfrak{T}$ such that $t(n) \geq d(n)$ for all $d \in \mathcal{R}_{mon}$ and all but finitely many n. Consequently, $\log t_{inv}$ is then a lower bound for the amount of nonconstructivity needed to learn $\mathcal{R}$ in the limit.

We continue by asking what amount of nonconstructivity is needed to obtain φ-minimal identification in the limit of the class $\mathcal{R}$. Now, the situation is intuitively more complex, since $\mathcal{LIM}_\varphi \setminus \mathcal{MIN}_\varphi \neq \emptyset$ for every $\varphi \in$ *Göd*. Interestingly, there are even Gödel numberings φ such that $\mathcal{MIN}_\varphi$ contains only classes of finite cardinality (cf. Freivalds [9]). On the other hand, the sufficient amount of nonconstructivity given in Theorem 2 does *not* depend on the Gödel numbering. Theorem 2 below is not the best possible and we shall improve it below, but it shows an easy way to achieve φ-minimal learning of of the class $\mathcal{R}$ in the limit.

THEOREM 2. *Let $\varphi \in$ Göd be arbitrarily fixed. Then there is a strategy $S \in \mathcal{P}^2$ such that the class $\mathcal{R}$ can be φ-minimal identified with nonconstructivity $n+1$ in the limit with respect to φ.*

Proof. Let $\varphi \in$ *Göd* be arbitrarily fixed, and let $n \in \mathbb{N}$. The help-word w is a bitstring b of length $n+1$ defined as follows. If $\varphi_i \in \mathcal{R}$, then the ith entry of b is 1, and 0 otherwise. So, the length of the help-word directly allows the strategy to compute n.

Next, assume any function $f \in \mathcal{R} \cap \{\psi_0, \psi_1, \ldots, \psi_n\}$, and let f^m and w be the input to S. Then S only considers those functions φ_i, $0 \leq i \leq n$, for which the ith entry in the help-word is 1. Since all these remaining functions are total, the strategy searches for the least index j among these functions for which $\varphi_j^m = f^m$. That is, it essentially uses the identification by enumeration principle (cf. Gold [12]). ∎

The proof of Theorem 2 looks quite simple which may be an indication that a smaller amount of nonconstructivity may suffice. Unfortunately, so far we could not show a lower bound for the amount of nonconstructivity needed to achieve φ-minimal learning in the limit of the class $\mathcal{R}$. On the other hand, as Theorem 4 below shows, we can achieve a much better result when allowing nonconstructivity $n+1$. This again indicates that we have used a too great amount of nonconstructivity in Theorem 2. And indeed, we can do exponentially better.

THEOREM 3. *Let $\varphi \in$ Göd be arbitrarily fixed. Then there is a strategy $S \in \mathcal{P}^2$ such that the class $\mathcal{R}$ can be φ-minimal identified with nonconstructivity $2 \cdot \log n$ in the limit with respect to φ.*

Proof. The key observation to show the theorem is that it suffices to know the *number* of recursive functions in the set $\{\varphi_0, \ldots, \varphi_n\}$. In order to use this information appropriately, the first half of the help-word is the binary encoding of n and the second half of the help-word w is just providing the number, say k, of recursive functions in the set $\{\varphi_0, \ldots, \varphi_n\}$. This number is also written in binary but leading zeros are added in order to ensure that both parts of the help-word have the same length. Thus $2 \cdot \log n$ many bits suffice to represent the help-word.

Next, assume any function $f \in \mathcal{R} \cap \{\psi_0, \psi_1, \ldots, \psi_n\}$, and let f^m and w be the input to S. Then the strategy S, by dovetailing its computations, first tries to compute $\varphi_i(0), \ldots, \varphi_i(m)$ for all $0 \le i \le n$ until it finds the first k programs $i_1, \ldots, i_k$ such that $\varphi_i(0), \ldots, \varphi_i(m)$ turn out to be defined for every $i \in \{i_1, \ldots, i_k\}$. Once the strategy S has found these programs $i_1, \ldots, i_k$, it outputs the least program $i \in \{i_1, \ldots, i_k\}$ for which it verifies $\varphi_i^m = f^m$ provided there is such a program, and m otherwise.

Taking into account that there must be $n + 1 - k$ many programs j among the programs $0, \ldots, n$ such that $\varphi_j \in \mathcal{P} \setminus \mathcal{R}$, for each of these programs j there must be a smallest y_j such that $\varphi_j(y_j)\uparrow$. Let $y_{\max}$ be the maximum of all these y_j. Hence, as soon as $m \ge y_{\max}$, the strategy S must find precisely the programs $i_1, \ldots, i_k$ such that $\varphi_i \in \mathcal{R}$ for all $i \in \{i_1, \ldots, i_k\}$. By assumption, the target function f possesses a program i with $0 \le i \le n$, and thus for all $m \ge y_{\max}$, the strategy must output $\min_\varphi f$. ∎

Next, we provide the theorem already mentioned above which shows that with nonconstructivity $n + 1$ a much stronger result is possible.

THEOREM 4. *Let $\varphi \in Göd$ be arbitrarily fixed. Then there is a strategy $S \in \mathcal{P}^2$ such that the class $\mathcal{R}$ can be φ-minimal finitely identified with nonconstructivity $n+1$ with respect to φ.*

Proof. Let $\varphi \in Göd$ be arbitrarily fixed, and let $n \in \mathbb{N}$. The help-word w is a bitstring b of length $n + 1$ defined as follows. If $\varphi_i \in \mathcal{R}$ and $\varphi_i \neq \varphi_j$ for all $0 \le j < i$, then the ith entry of b is 1, and 0 otherwise. So, the length of the help-word directly allows the strategy to compute n.

Note that now the help-word allows for implicitly having a one-to-one enumeration for the functions $f \in \mathcal{R} \cap \{\varphi_0, \ldots, \varphi_n\}$.

Next, assume any function $f \in \mathcal{R} \cap \{\varphi_0, \varphi_1, \ldots, \varphi_n\}$, and let f^m and w be the input to S. Then S only considers those functions φ_i, $0 \le i \le n$, for which the ith entry in the help-word is 1.

For all these i, the strategy computes φ_i^m and checks whether or not they are pairwise different. As long as this is not the case, the strategy outputs m. If all these φ_i^m are pairwise different, then the strategy outputs the i for which it could verify $f^m = \varphi_i^m$.

By construction, it is obvious that S finitely converges to $\min_\varphi f$. ∎

Again, we still could not prove the amount of nonconstructivity given in Theorem 4 to be the best possible.

We therefore continue to look at the case, where we have to learn an indexed family $\mathcal{U}$ of recursive functions. Note that for every indexed family $\mathcal{U}$ and any of its numberings $\psi \in \mathcal{R}^2$ we have $\mathcal{U} \in \mathcal{MIN}_\psi$ (cf. Gold [12]). In contrast, $\mathcal{NUM} \# \mathcal{FIN}$ (see e.g., [23] and the references therein). So, it is only natural to ask for the amount of nonconstructivity needed to finitely learn ψ-minimal programs. The answer is provided by our theorems below.

Theorem 5. *Let $\mathcal{U}$ be any indexed family, and let $\psi \in \mathcal{R}^2$ be any numbering for $\mathcal{U}$. Then there is a strategy $S \in \mathcal{P}^2$ such that the class $\mathcal{U}$ can be ψ-minimal finitely identified with nonconstructivity $2 \cdot \log n$ with respect to ψ.*

Proof. The key observation for the proof is that it suffices to know the number k of distinct functions in $\{\psi_0, \ldots, \psi_n\}$. Thus, the help-word w is again divided in two halves, where the first half is the binary representation of n and the second half provides the number k (again including leading zeros). Thus $2 \cdot \log n$ many bits suffice for representing the help-word.

On input f^m and w the strategy S computes, by dovetailing its computations, $\psi_i(x)$ for all $i \in \{0, \ldots, n\}$ and $x = 0, 1, 2, \ldots$ until it has verified that there are exactly k different functions. Let $i_1, \ldots, i_k$ be the least indices of these k different functions. Next, it checks whether or not there is precisely one $i \in \{i_1, \ldots, i_k\}$ such that $f^m = \psi_i^m$. If this is the case, the strategy outputs this i. Otherwise, it outputs m.

By construction, it is obvious that S finitely converges to $\min_\psi f$. ∎

Next we show that the amount of nonconstructivity given in Theorem 5 cannot be substantially reduced.

Theorem 6. *There is an indexed family $\mathcal{U}$ and a numbering $\psi \in \mathcal{R}^2$ for it such that no strategy $S \in \mathcal{P}^2$ can ψ-minimal finitely identify the class $\mathcal{U}$ with nonconstructivity $c \cdot \log n$ with respect to ψ, where $c \in (0, 1)$ is any constant.*

Proof. We construct the indexed family $\mathcal{U}$ by defining the numbering $\psi \in \mathcal{R}^2$ for it. For this purpose, we use the following pairing function $c\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, where $c(x, y) = 2^x(2y+1)-1$. Note that this pairing function is a bijection. It may be traced back to Pepis [19] and Kalmár [14]. Furthermore, we interpret every function in $\mathcal{P}^2$ as a strategy and obtain thus an effective enumeration S_0, S_1, S_2, ... of all possible strategies. Below, for $\ell \in \mathbb{N}$, we use the shortcut $i^{\ell+1}$ to denote the encoding f^ℓ of the initial segment of the function f for which $f(z) = i$ for all $i = 0, \ldots, \ell$.

For every $i \in \mathbb{N}$ we define two functions ψ_{2i} and ψ_{2i+1} as follows. Let x and y be the uniquely determined numbers such that $i = c(x, y)$. Now, we successively define for $k = 1, 2, 3, \ldots$ the functions values $\psi_{2i}(k-1) = \psi_{2i+1}(k-1) = i$ and input i^k and y to the strategy S_x until we find the smallest k such that the following Conditions (A) and (B) are satisfied.

(A) There is an $\ell < k$ such that each of the values $S_x(i, y), \ldots, S_x(i^{\ell+1}, y)$ turns out to be computable in at most k steps.

(B) $S_x(i, y) \neq S_x(i^2, y) \neq \cdots \neq S_x(i^\ell, y) = S_x(i^{\ell+1}, y)$.

If Conditions (A) and (B) never turn out to be satisfied then the function values $\psi_{2i}(k)$ and $\psi_{2i+1}(k)$ are defined for all $k \in \mathbb{N}$, and thus $\psi_{2i}, \psi_{2i+1} \in \mathcal{R}$.

On the other hand, if Conditions (A) and (B) turn out to be satisfied then Condition (B) implies that the sequence $S_x(i, y), \ldots, S_x(i^{\ell+1}, y)$ tends to converge finitely. That is, it either converges finitely or it cannot converge finitely at all. Now, we continue to define the functions ψ_{2i} and ψ_{2i+1} as follows.

(C) If $S_x(i^\ell, y) = 2i$, then we define $\psi_{2i}(z) = i + k + z$ for all $z \geq k$.

Furthermore, we set $\psi_{2i+1}(z) = i$ for all $z \geq k$.

(D) If $S_x(i^\ell, y) = 2i + 1$, then we define $\psi_{2i+1}(z) = i + k + z$ for all $z \geq k$.

Furthermore, we set $\psi_{2i}(z) = i$ for all $z \geq k$.

(E) If $S_x(i^\ell, y) \notin \{2i, 2i + 1\}$, then we define $\psi_{2i}(z) = \psi_{2i+1}(z) = z$ for all $z \geq k$.

So again we obtain that $\psi_{2i}, \psi_{2i+1} \in \mathcal{R}$, and consequently, $\psi \in \mathcal{R}^2$. Finally, we set the desired class $\mathcal{U} = \mathcal{R}_\psi$.

It remains to show that there is no strategy $S \in \mathcal{P}^2$ that ψ-minimal finitely identifies $\mathcal{U}$ with nonconstructivity $c \cdot \log n$ with respect to ψ, where $c \in (0, 1)$ is any constant.

Suppose the converse, i.e., there is a strategy $S \in \mathcal{P}^2$ that ψ-minimal finitely identifies $\mathcal{U}$ with nonconstructivity $c \cdot \log n$ with respect to ψ. Then there must be a $v \in \mathbb{N}$ such that $S = S_v$ in our enumeration $S_0, S_1, S_2, \ldots$ of all possible strategies. Let d be the function from Definition 4. Furthermore, for every $n \in \mathbb{N}$ and every $f \in \{\psi_0, \ldots, \psi_n\}$ there has to be a help-word w of length at most $d(n)$ and depending only on n such that the sequence $(S_v(f^m, w))_{m \in \mathbb{N}}$ finitely converges to the minimal ψ-program of f.

By assumption, we know that there is a $c \in (0, 1)$ such that $d(n) \leq c \cdot \log n$. Hence, for n large enough we conclude that $d(n) > 1$ and $(1-c)/2 > (2+(v+1))/\log n$.

$$\begin{array}{rcl} 1 & > & \dfrac{d(n)}{\log n} + \dfrac{2+(v+1)}{\log n} \\ \log n & > & d(n) + 2 + (v+1) \\ \log n - \log 2^{v+1} & > & d(n) + 2 \\ \dfrac{n}{2^{v+1}} & > & 2^{d(n)+2} \\ \dfrac{n+2}{2^{v+1}} & > & 2 \cdot 2^{d(n)} + 1 \\ \dfrac{n+2}{2^{v+1}} & > & 2w + 1 \ , \quad \text{since } w \le 2^{d(n)} \\ \dfrac{n}{2} & > & 2^v(2w+1) - 1 \\ \dfrac{n}{2} & > & c(v,w) \ . \end{array}$$

Now, let $i = c(v,w)$ and we consider the functions ψ_{2i} and ψ_{2i+1}. By our choice of n, these functions must be among the functions $\{\psi_0, \ldots, \psi_n\}$.

Let $\ell \in \mathbb{N}^+$ be the least number such that S_v on two successive inputs outputs the same hypothesis, i.e., $S_v(i,w) \neq \cdots \neq S_v(i^\ell, w) = S_v(i^{\ell+1}, w)$. Such an ℓ has to exist, since otherwise S_v can neither finitely identify ψ_{2i} nor ψ_{2i+1}.

If $S_v(i^\ell, w) \notin \{2i, 2i+1\}$ we are already done, since ψ_{2i} and ψ_{2i+1} are the only functions from $\mathcal{U}$ having an initial segment where all values are equal to i.

Finally, if $S_v(i^\ell, w) \in \{2i, 2i+1\}$ then by construction (cf. Condition (C) and (D), respectively, above) we know that $\psi_{2i}(z) = \psi_{2i+1}(z)$ for all $z = 0, \ldots, \ell$ but clearly $\psi_{2i} \neq \psi_{2i+1}$. Thus, the strategy S_v fails to identify finitely either function ψ_{2i} or function ψ_{2i+1}. ∎

As the proof of Theorem 6 shows, the failure to ψ-minimal finitely identify the indexed family $\mathcal{U}$ with respect to the numbering ψ with nonconstructivity $c \cdot \log n$, for $c \in (0,1)$, is caused by the requirement to finitely identify the functions from $\mathcal{U}$. Thus, we directly obtain the following corollary.

Corollary 7. *There is an indexed family $\mathcal{U}$ and a numbering $\psi \in \mathcal{R}^2$ for it such that no strategy $S \in \mathcal{P}^2$ can finitely identify the class $\mathcal{U}$ with nonconstructivity $c \cdot \log n$ with respect to ψ, where $c \in (0,1)$ is any constant.*

4. Conclusions

We have presented a model for the inductive inference of recursive functions that incorporates a certain amount of nonconstructivity. In our model, the amount of nonconstructivity needed to solve the learning problems considered has been used as a quantitative characterization of their difficulty.

We studied the problem of learning the whole class $\mathcal{R}$ under various postulates. These postulates range from learning in the limit to finite and minimal identification.

As far as learning in the limit is concerned, the amount of nonconstructivity needed to learn $\mathcal{R}$ can be very small and there is no smallest amount that can be described in a computable way (cf. Theorem 1).

This result is nicely contrasted by the fact that we needed nonconstructivity $2 \cdot \log n$ to φ-minimal identify the class $\mathcal{R}$ in the limit and nonconstructivity $n+1$ to φ-minimal finitely identify $\mathcal{R}$ (cf. Theorems 3 and 4, respectively). That is, each additional postulate exponentially increased the amount of nonconstructivity needed. It remains, however, open whether or not these results can be improved.

Furthermore, we investigated the amount of nonconstructivity needed to φ-minimal finitely identify any indexed family of recursive functions. In this setting we obtained an upper bound of $2 \cdot \log n$ for the amount of nonconstructivity needed and showed that this amount cannot be substantially improved (cf. Theorems 5 and 6).

References

[1] Leonard M. Adleman and M. Blum. Inductive inference and unsolvability. *The Journal of Symbolic Logic*, 56(3):891–900, 1991.

[2] Jānis M. Bārzdiņš. Complexity of programs to determine whether natural numbers not greater than n belong to a recursively enumerable set. *Soviet Mathematics Doklady*, 9:1251–1254, 1968.

[3] Manuel Blum. A machine independent theory of the complexity of recursive functions. *Journal of the ACM*, 14(2):322–336, 1967.

[4] John Case and Carl Smith. Comparison of identification criteria for machine inductive inference. *Theoret. Comput. Sci.*, 25(2):193–220, 1983.

[5] Peter Cholak, Rod Downey, Lance Fortnow, William Gasarch, Efim Kinber, Martin Kummer, Stuart Kurtz, and Theodore A. Slaman. Degrees of inferability. In *Proc. 5th Annual ACM Workshop on Comput. Learning Theory*, pages 180–192, New York, 1992. ACM Press.

[6] Paul Erdős. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53(4):292–294, 1947.

[7] R. V. Freivald and R. Wiehagen. Inductive inference with additional information. *Elektronische Informationsverarbeitung und Kybernetik*, 15(4):179–185, 1979.

[8] R.V. Freivald. Minimal gödel numbers and their identification in the limit. In Jiří Bečvář, editor, *Mathematical Foundations of Computer Science 1975, 4th Symposium, Mariánské Lázně, September 1-5, 1975*, volume 32 of *Lecture Notes in Computer Science*, pages 219–225, Berlin, 1975. Springer-Verlag.

[9] Rūsiņš Freivalds. Inductive inference of recursive functions: Qualitative theory. In J. Bārzdiņš and D. Bjørner, editors, *Baltic Computer Science*, volume 502 of *Lecture Notes in Computer Science*, pages 77–110. Springer, Berlin, 1991.

[10] Rūsiņš Freivalds. Amount of nonconstructivity in finite automata. In Sebastian Maneth, editor, *Implementation and Application of Automata, 14th International Conference, CIAA 2009, Sydney, Australia, July 14-17, 2009. Proceedings*, volume 5642 of *Lecture Notes in Computer Science*, pages 227–236, Berlin, 2009. Springer.

[11] E. Mark Gold. Limiting recursion. *Journal of Symbolic Logic*, 30:28–48, 1965.

[12] E. Mark Gold. Language identification in the limit. *Inform. Control*, 10(5):447–474, 1967.

[13] Sanjay Jain, Daniel Osherson, James S. Royer, and Arun Sharma. *Systems that Learn: An Introduction to Learning Theory, second edition.* MIT Press, Cambridge, Massachusetts, 1999.

[14] László Kalmár. On the reduction of the decision problem. First Paper. Ackermann prefix, a single binary predicate. *The Journal of Symbolic Logic*, 4(1):1–9, 1939.

[15] Richard M. Karp and Richard J. Lipton. Turing machines that take advice. *L' Enseignement Mathématique*, 28:191–209, 1982.

[16] Е.Б Кинбер. О предельном синтезе почти минимальных геделевских номеров. In Jānis Bārzdiņš, editor, Теория Алгоритмов и Программ, volume I, pages 212–223. Latvian State University, 1974.

[17] Martin Kummer and Frank Stephan. On the structure of the degrees of inferability. *J. of Comput. Syst. Sci.*, 52(2):214–238, 1996.

[18] L. H. Landweber and E. L. Robertson. Recursive properties of abstract complexity classes. *Journal of the ACM*, 19(2):296–308, 1972.

[19] Joséf Pepis. Ein Verfahren der mathematischen Logik. *The Journal of Symbolic Logic*, 3(2):61–76, 1938.

[20] Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967. Reprinted, MIT Press 1987.

[21] B. A. Trakhtenbrot and Ya. M. Barzdin. *Finite Automata, Behavior and Synthesis.* North Holland, Amsterdam, 1973.

[22] Rolf Wiehagen. *Zur Theorie der Algorithmischen Erkennung.* Dissertation B, Humboldt-Universität zu Berlin, 1978.

[23] Thomas Zeugmann and Sandra Zilles. Learning recursive functions: A survey. *Theoret. Comput. Sci.*, 397(1-3):4–56, 2008.