Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

Randomization, Martingales and Additional Information in Inductive Inference *

Rūsiņš Freivalds   Marek Karpinski
Carl H. Smith†

Institute of Math and Computer Science
University of Latvia
Raiņa bulvāris 29,LV-1459
Riga, Latvia
rusins@cclu.lv

Department of Computer Science
University of Bonn
53117, Bonn
Germany
karpinski@cs.bonn.edu

Institute for Logic, Language and Computation
University of Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
The Netherlands
csmith@fwi.uva.nl

Abstract

Finite identification, sometimes called "one shot learning," is the most basic identification type studied in Inductive Inference. There are several ways to generalize this notion. One of the more popular generalizations is to consider identification in the limit, as opposed to one attempt. We other generalization that we consider are randomized finite identification and finite identification with an additional information. We prove that these lines of generalization are not merely independent (neither one majorize the other one) but also incompatible (simultaneous generalization into two of these directions provides no generalization at all).

1 Introduction

Inductive Inference is the term used for the synthesis of programs from sample computations. It is the part of computational learning theory based on recursion theory. Started by the well-known paper [Go 67] nowadays inductive inference is the most developed part of computational learning theory (see the survey [AS 83] and the monograph [OSW 86]). Although far from development of machine learning algorithms, inductive inference has actively supplied practitioners with intuitions on how to learn, what to learn, and what difficulties to avoid.

The main technique of research in Inductive Inference is the set theoretic comparison of various identification types and finding deeper relations among the types and among the identifiable classes of functions and languages. The types are chosen so as to highlight particular aspects of the learning process, such as compairing probabilistic learning to deterministic learning.

In this paper, we consider two different generalizations of the most fundamental of the learning by example models (one shot learning). One generalization uses extra information, like an upper bound on the size of the solution, and the other uses randomized learning proceedures. Not surprisingly, we

*This project was supported by Latvian Science Council Grant No. 93.599

†On leave from the Department of Computer Science, University of Maryland, College Park, MD 20742 USA.

find that these two avenues of generalization are completely independent. What is surprising is that while each of the two techniques does turn out to generalize the one shot model independently, when the two techniques are combined so as to considered randomized one shot learning with extra information, then there is absolutely NO generalization at all. Most of the paper is concerned with the precise set theoretic compairisons that arise by considering the collections of sets that are learnable with respect to each of the generalizations considered. There are many definitions and proofs needed to make the above notions precise, but this is the nature of such a mathematical study. The individual theorems by themselves mean very little. Most of the proofs are omitted due to space considerations. The contribution of this paper comes from the combination of the results and the creation of Figure 1.

We proceed to give the basic notation needed to precisely describe the identification types we consider in this paper.

Let $\mathcal{P}, \mathcal{R}$ denote the sets of all partial recursive and total recursive functions of one argument taken from the natural numbers. Suppose L and M are sets. Then $L \subseteq M$ denotes the inclusion of L in M, and $L \subset M$ denotes the proper inclusion of the same sets.

Let φ be a fixed Gödel, numbering or acceptable programming system, of all the partial recursive functions of one argument. For any $f \in P$ let $\min_\varphi f$ denote the minimal index of f in φ. If $f(x)$ is defined for all $x \leq n$ then $f^{[n]}$ denotes a finite string consisting of integers $(f(0), f(1), \ldots, f(n))$. We say that the sequence of integers $(x_n)_{n \in N}$ converges to x ($x = \lim_n x_n$) if there exists $n_0 \in N$ such that $x_n = x$ for all $n > n_0$. Finally, let $id(x) = x$ and $id^2(x) = x^2$ for any $x \in N$.

The most widely studied inference types are called FIN (one shot learning), EX (learning in the limit), BC (semantic, rather than syntactic, convergence). The definitions given below are equivalent to the standard ones. Our choice of notation here is to facilitate the comparisons we make later on. All undefined concepts and notation can be found in [OSW 86] and [Smi 94].

Definition 1.1 *Let $\mathcal{U}$ be a class of total recursive functions and φ be a Gödel numbering of all 1-argument partial recursive functions. $\mathcal{U}$ is called finitely identifiable ($\mathcal{U} \in FIN$) if and only if there is a recursive functional F on $\mathcal{U}$ such that for arbitrary $f \in \mathcal{U}$,*

$$\varphi_{F(f)} = f$$

Inductive inference with extra information has also been investigated. One particularly useful piece of extra information is an upper bound on the index of a program for the function being learned. Consequently, we make the following definition.

Definition 1.2 *Let $\mathcal{U}$ be a class of total recursive functions and φ be a Gödel numbering of all 1-argument partial recursive functions. $\mathcal{U}$ is called finitely identifiable with extra information ($\mathcal{U} \in FIN^+$) if and only if there is a recursive functional F on $\mathcal{U}$ such that for arbitrary $f \in \mathcal{U}$ and any $b \geq \min_\varphi f$*

$$\varphi_{F(b,f)} = f.$$

Studying the inductive inference of minimal programs, R. Freivalds [Fre 78] found that a certain criterion becomes simple and natural if one considers counterparts for FIN (where the recursive functional is replaced by an effective operation), and (Definition 1.5 below) EX (where the strategy working in the limit is replaced by a limit-effective operation).

Definition 1.3 *A functional F on $\mathcal{U}$ (of type $\mathcal{U} \to N$) is called an* effective operation *if there is a partial recursive function ψ such that for arbitrary $\varphi_x \in \mathcal{U}$:*

1. *$F(\varphi_x)$ defined $\Leftrightarrow \psi(x)$ defined,*
2. *$F(\varphi_x)$ defined $\Rightarrow (F(\varphi_x) = \psi(x))$.*

Definition 1.4 *A class $\mathcal{U}$ of total recursive functions is called* finitely standardizable *($\mathcal{U} \in FSTAND$) if and only if there is an effective operation F on $\mathcal{U}$ such that for arbitrary $f \in \mathcal{U}$*

$$\varphi_{F(f)} = f.$$

It is easy to see that $\mathcal{U} \in FIN$ implies $\mathcal{U} \in FSTAND$. There is a classical theorem suggesting the converse might also be true.

Theorem 1 Kreisel, Lacombe, Shoenfield [KLS 57] *F is a total effective operation on $\mathcal{R}$ if and only if F is a total recursive functional on $\mathcal{R}$.*

Since standardizability is performed by a total effective operation, one might suggest that $FSTAND = FIN$. However, $FSTAND \neq FIN$ proved in [FKW 84]. Since then, the notion of standardizability has been studied by many authors (see [OSW 86]) where it was believed that $FSTAND$ is only "slightly" larger than FIN. We show below that this belief is not accurate.

Recursive functionals are computed by Turing machines that receive the graph of a function f on the input tape, and produce the result $F(f)$ (if it is defined) in a finite number of steps. However, if the value of the functional $F(f)$ is not defined, then the machine can go on working indefinitely. Since this definition has already involved the potentially infinite processing of the input information, it was natural for E. M. Gold [Go 67] to develope a genuinely infinite identification process.

Definition 1.5 *Let $\mathcal{U}$ be a class of total recursive functions and φ be a Gödel numbering of all 1-argument partial recursive functions. $\mathcal{U}$ is called* identifiable in the limit *($\mathcal{U} \in EX$) if and only if there is an algorithmic device that takes as input the graph of a function f and produces outputs converging to an integer i such that*

$$\varphi_i = f.$$

Definition 1.6 *Suppose ($\mathcal{U} \in EX$) as witnessed by an algorithmic device M. If, for any $f \in \mathcal{U}$, M changes its output no more than n times enroute to convergence (we say that the machine makes no more than n mindchanges) then $\mathcal{U}$ is in EX_n.*

Definition 1.7 *Let $\mathcal{U}$ be a class of total recursive functions and φ be a Gödel numbering of all 1-argument partial recursive functions. $\mathcal{U}$ is called BC-*identifiable *if and only if there is an identification machine such that given the graph of an arbitrary $f \in \mathcal{U}$, it produces an infinite sequence of outputs $i_0, i_1, \cdots$ such that*

$$\varphi_{i_t} = f$$

for all but finitely many t.

Randomized Turing machines are deterministic Turing machines having access to the simplest Bernoulli random number generators equiprobably outputting zeros and ones. Informally, one can say that the machine can toss coins.

Definition 1.8 *Let $\mathcal{U}$ be a class of total recursive functions and φ be a Gödel numbering of all 1-argument partial recursive functions. $\mathcal{U}$ is said to be in $randFIN(p)$ if and only if there is a randomized identification machine such that given the graph of an arbitrary $f \in \mathcal{U}$, it produces an output i such that*

$$\varphi_i = f$$

with probability no less than p.

Please notice that every recursive function has infinitely many indices in every Gödel numbering. Hence it is possible that every particular integer is output with very small probability but the total of the probabilities for correct programs of the target function exceeds p.

It is well known that $EX \subset BC$. The properness of the inclusion was proved by J. Barzdins ([Bar 74], see [OSW 86]). It can be easily seen from the definitions that

$$FIN \subseteq FSTAND \subseteq FIN^+$$

and

$$FIN \subseteq randFIN.$$

These generalizations of FIN really expand the capabilities of the identification machines. We have proved that these generalizations are independent in the sense that the capabilities of one generalization do not majorize the capabilities of the other generalization. The inclusions we find are summarized in the following diagram.

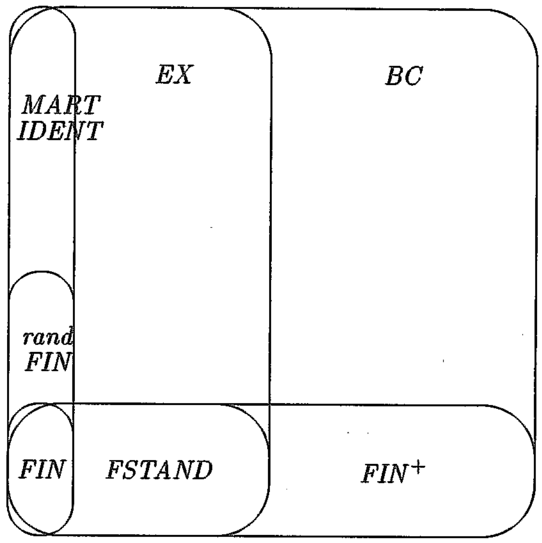

Figure 1. Summary of Results

What we find surprising, is that these two lines of generalization are incompatible. If we simultaneously generalize the notion of FIN-identifiability along these two generalization lines, then the capabilities of the identification machines do not increase at all (Theorem 13).

2 Randomized finite identification and martingales

We continue by reviewing the pertinant previous results concerning the learnable classes defined in the previous section.

Theorem 2 (R. Freivalds [Fre 79]) *If $p > 2/3$, then $randFIN(p) = FIN$.*

Theorem 3 **(R.Freivalds [Fre 79])**
If, for real numbers p_1 and p_2, there is an integer n such that $\frac{n+2}{2n+3} \le p_1 \le p_2 < \frac{n+1}{2n+1}$, then $randFIN(p_1) = randFIN(p_2)$.

Theorem 4 **(R.Freivalds [Fre 79])**
If, for real numbers p_1 and p_2, there is an integer n such that $p_1 \le \frac{n+1}{2n+1}$ and $p_2 \ge \frac{n+1}{2n+1}$, then $randFIN(p_1) \ne randFIN(p_2)$.

The following lemma was implicitly present in [Fre 79], however it was not singled out from the proof of Theorem 3 in [Fre 79].

Lemma 2.1 *If a class $\mathcal{U}$ of total recursive functions is in $randFIN(p)$ with $p > \frac{n+2}{2n+3}$, then $\mathcal{U}$ is finitely identified by a team of $2n+1$ deterministic identification machines such that every function in $\mathcal{U}$ is correctly identified by at least $n+1$ machines of this team.*

Theorem 5 *If a class $\mathcal{U}$ of total recursive functions is in $randFIN(\frac{n+1}{2n+1})$ then $\mathcal{U}$ is identifiable with at most n mindchanges ($\mathcal{U} \in EX_n$).*

Proof. Since $\frac{n+1}{2n+1} > \frac{n+2}{2n+3}$, it follows from Lemma 2.1 that there exists a team of $2n+1$ deterministic identification machines such that at least $n+1$ machines of this team correctly finitely identify the class $\mathcal{U}$. Our limit identification machine simulates the machines in this team. The machine waits until the first bunch of at least $n+1$ outputs is obtained from the machines to be simulated. Then our machine outputs a result being the amalgamation of the results of the simulated machines. When some of these outputs turn out to have produced a wrong result, our machine waits until a new bunch of $n+1$ outputs not yet found incorrect is found, and makes amalgamation of these results. Clearly, such change can happen at most n times. □

Theorem 6 *For arbitrary positive integer n there is a class $\mathcal{U}$ of total recursive functions such that:*

1. $\mathcal{U} \in randFIN(\frac{n+1}{2n+1})$,
2. $\mathcal{U} \notin EX_{n-1}$.

Theorem 7 *There is a class $\mathcal{U}$ of total recursive functions such that $\mathcal{U}$ is identifiable with at most 1 mindchange but $\mathcal{U}$ is not in $randFIN$.*

Proof. The class $\mathcal{U}$ consists of the "constant zero" function and all functions differring from it on exactly one argument. Assume by way of contradiction that there is a randomized finite identification machine identifying the class $\mathcal{U}$ with probability $> \frac{1}{2} + \frac{1}{n}$. Consider a sufficiently long initial segment fr of the constant function such that the machine outputs some results with total probability $> \frac{1}{2} + \frac{1}{n}$ using only input from fr.

We noneffectively choose a function $g \in \mathcal{U}$ such that the machine has output correct programs for g with a probability at most $\frac{1}{2n}$ on fr. Since $\mathcal{U}$ contains infinitely many functions, all the functions cannot have a probability over $\frac{1}{2n}$ on f. The machine has already used the probability $\frac{1}{2} + \frac{1}{n}$ on f. For the new outputs there remains probability less than $\frac{1}{2} - \frac{1}{n}$ which is not enough for the function g. □

Now we consider a generalization of randomized finite identification. We wish to point out a specified property of $randFIN$-identifiability. For this identification type it is specific that at some finite moment (namely, when the total probability of the output results exceeds a definite level for the first time) the probabilities of sufficiently many outputs are already determined. Not much can be changed after this moment. This is similar to deterministic finite identification. However, some minor changes are still possible.

Randomness is closely related to the notion of measure. Many approaches to defining the notion of measure have been considered. Most of the modern definitions are based on the Lebesgue measure. From the viewpoint of classical mathematics, the class of all recursive functions has measure 0. However, based on the ideas of Lebesgue, measure has been introduced for 0-1 valued total recursive functions as well. There have been several (nearly equivalent) definitions. The most popular definition is due to K. Mehlhorn [Meh 73]. It uses the notion of martingales.

A function $m : \{0,1\}^* \to$ is a martingale if for all x, if $m(x0)\downarrow$ or $m(x1)\downarrow$ then

1. $m(x)\downarrow$, $m(x0)\downarrow$ and $m(x1)\downarrow$
2. $m(x) = \frac{m(x0)+m(x1)}{2}$

Let $f \in \{0,1\}^\omega$. We say m *wins on* f if

1. For all n, $m(f(0)\cdots f(n))\downarrow$, and
2. $\limsup_{n\to\infty} m(f(0)\cdots f(n)) = \infty$.

We say m *loses on* f if m does not win on f. Let $\mathcal{C}$ be a collection of functions. then m *wins on* $\mathcal{C}$ if m wins on every $f \in \mathcal{C}$. We say a class $\mathcal{C}$ has measure zero if there is a martingale m that wins on all $f \in \mathcal{C}$. We define recursive and partial recursive measure zero by requiring m to be recursive or partial recursive respectively.

Since we are interested in arbitrary total recursive functions, not only the 0,1-valued ones, we consider a modified notion of martingales. We consider Turing machines that monotonically limit-computing non-negative real valued functions. This means that the machine outputs newer and newer hypotheses about the result. Each hypothesis is a non-negative rational number, and the hypotheses never decrease.

Definition 2.1 *We say that a class $\mathcal{U}$ of total recursive functions has a martingale (denoted as $\mathcal{U} \in \mathcal{MART}$) if there is a deterministic machine receiving as the input the graph of the target function such that:*

1. *for every $f \in \mathcal{U}$, the machine outputs an initial segment $f^{[v(f)]}$ of the target function in a finite number of steps.*
2. *for every $f \in \mathcal{U}$, the machine monotonically limit-computes a real number $m(f^{v(f)})$.*
3. *for every $f \in \mathcal{U}$, for every $x > v(f)$, and for every initial segment of a total function g such that $f^{[v(f)]} = g^{[v(f)]}$, the machine monotonically limit-computes $m(g^{[x]})$ such that:*
   a) $0 \leq m(g^{[x+1]}) \leq m(g^{[x]})$,
   b) *the total of $m(h^{[x+1]})$ over all the functions h such that $h^{[x]} = g^{[x]}$, never exceeds $m(g^{[x]})$,*
   c) *for every $f \in \mathcal{U}$, there is a y such that $m(f^{[y]}) = m(f^{[y+1]}) = m(f^{[y+2]}) = \ldots > 0$.*

Proposition 2.1 *If a class $\mathcal{U}$ of total recursive functions is in $randFIN$, then $\mathcal{U}$ is in $MART$.*

Proof. The probabilities of corresponding output programs can be used for the $m(f^{[x]})$. □

Definition 2.2 *We say that a class $\mathcal{U}$ of total recursive functions has an* accumulation point *if there is a function $f \in \mathcal{U}$ such that for arbitrary initial segment $f^{[x]}$ of the function f there is a different function $g \in \mathcal{U}$ with the same initial segment $g^{[x]} = f^{[x]}$.*

Definition 2.3 *We say that a class of total recursive functions is* discrete *if the class has no accumulation points.*

Proposition 2.2 *If a class $\mathcal{U}$ of total recursive functions is in $MART$, then $\mathcal{U}$ is discrete.*

Proof. The proof follows from the requirement 3c in the definition of the martingale. □

$\mathcal{U} \in \mathcal{MART}$ means that the class $\mathcal{U}$ is small in measure. However, this notion is rather distant from notions of learnability. Hence we consider an identifiability-related counterpart of this notion.

Definition 2.4 *We say that a class $\mathcal{U}$ of total recursive functions is a martingale identifiable (denoted as $\mathcal{U} \in \mathcal{MARTIDENT}$) if there is a deterministic machine receiving as the input the graph of the target function such that:*

1. *for every $f \in \mathcal{U}$, the machine outputs an initial segment $f^{[v(f)]}$ of the target function and a program consistent with this initial segment in a finite number of steps.*
2. *for every $f \in \mathcal{U}$, the machine monotonically limit-computes a real number $m(f^{v(f)})$.*
3. *for every $f \in \mathcal{U}$, for every $x > v(f)$, and for every initial segment of a total function g such that $f^{[v(f)]} = g^{[v(f)]}$, the machine monotonically limit-computes $m(g^{[x]})$ such that:*
   a) $0 \leq m(g^{[x+1]}) \leq m(g^{[x]})$,
   b) *the total of $m(h^{[x+1]})$ over all the functions h such that $h^{[x]} = g^{[x]}$, never exceeds $m(g^{[x]})$,*
   c) *for every $f \in \mathcal{U}$, there is a y such that $m(f^{[y]}) = m(f^{[y+1]}) = m(f^{[y+2]}) = \ldots > 0$.*
4. *at every moment when the machine outputs the first positive approximation for a value of the martingale function on an initial segment of some function, the machines produces a program consistent with this initial segment of the target function,*
5. *for every $f \in \mathcal{U}$, there is a y such that the program output on this initial segment is the same as the program output on the preceding initial segment of the same function.*

Proposition 2.3 *If a class $\mathcal{U}$ of total recursive functions is in $randFIN$, then $\mathcal{U}$ is in $MARTIDENT$.*

Proof. The probabilities of corresponding output programs can be used for the $m(f^{[x]})$. □

3 Identification with an additional information

Theorem 8 *There is a class $\mathcal{U}$ of total recursive functions such that:*

1. $\mathcal{U} \in FIN^+$,
2. $\mathcal{U} \notin EX$.

Theorem 9 *If a class $\mathcal{U}$ of total recursive functions is finitely standardizable, then $\mathcal{U}$ is in EX.*

Theorem 10 *There is a class $\mathcal{U}$ of total recursive functions such that:*

1. *$\mathcal{U}$ is finitely standardizable,*

2. *$\mathcal{U}$ is not identifiable in the limit with any constant number of mindchanges.*

Proof. A non-empty class $\mathcal{U}$ of total recursive functions and effective operation on $\mathcal{U}$ will be constructed such that:

a) for arbitrary $f \in \mathcal{U}$, the value of $F(f)$ is a correct φ-index of the function f,

b) for arbitrary initial segment $f^{[x]}$ of an arbitrary function $f \in \mathcal{U}$ there is a different function $g \in \mathcal{U}$ with the same initial segment $g^{[x]} = f^{[x]}$.

1. The assertion 1) of our Theorem is implied by a) and b).

2. We construct the class $\mathcal{U}$ and the effective operation F with the properties a), b). Every function in the class $\mathcal{U}$ is a function differing from some constant only on a finite number of arguments.

We construct simultaneously the class $\mathcal{U}$ and a recursively enumerable set T of pairs $\langle a, b\rangle$ (to be used for the definition of the effective operation). In the process of the construction some functions will be placed into an auxiliary class $\mathcal{U}'$. If a function gets into $\mathcal{U}'$, it can be removed from $\mathcal{U}'$ or it can stay in $\mathcal{U}'$ forever. If a function is removed, it never returns to $\mathcal{U}'$. The class $\mathcal{U}$ consists of all the functions such that they stay in $\mathcal{U}'$ forever. By $a_0a_1\ldots a_n\alpha^\infty$ we denote the function

$$f(x) = \begin{cases} a_0, & \text{if } x = 0; \\ a_1, & \text{if } x = 1; \\ \cdots & \\ a_n, & \text{if } x = n; \\ \alpha, & \text{if } x > n. \end{cases}$$

By $\{a_0a_1\ldots a_n\}$ we denote the canonical index of the string $a_0a_1\ldots a_n$. By $s(a_0a_1\ldots a_n)$ we denote a φ-index of the function

$$g(x) = \begin{cases} a_0, & \text{if } x = 0; \\ a_1, & \text{if } x = 1; \\ \cdots & \\ a_n, & \text{if } x = n; \\ a_n, & \text{if } x > n. \end{cases}$$

obtained from $\{a_0a_1\ldots a_n\}$ by usage of some fixed uniform procedure. The construction of $\mathcal{U}$ and T is organized in stages.

Stage 0. The functions 0^∞ and 1^∞ are placed in $\mathcal{U}'$ and the pairs $\langle\{0\}, 0\rangle$, $\langle\{1\}, 0\rangle$ are placed into T. Go to Stage 1.

Stage $n+1$. Assume by induction that every pair $\langle a, b\rangle$ placed into T at Stage n is of form $\langle\{a_0a_1\ldots a_n\}, n\rangle$ and it corresponds to a function $a_0a_1\ldots a_n a_n^\infty$ placed into $\mathcal{U}'$. We also assume by induction that at the end of Stage n $\mathcal{U}'$ does not contain any functions different from those corresponding to pairs placed into T at Stage n.

We compute n steps of each of the following computations:

$$\begin{array}{llll} \varphi_0(0) & & & \\ \varphi_1(0), & \varphi_1(1) & & \\ \cdots & & & \\ \varphi_n(0), & \varphi_n(1), & \ldots, & \varphi_n(n) \end{array}$$

After that we consider all the pairs placed into $\mathcal{U}'$ at Stage n. When considering the pair $\langle\{a_0a_1\ldots a_n\}, n\rangle$ we test whether or not there is a $k \le n$ such that:

(i) computation of each of $\varphi_k(0)$, $\varphi_k(1), \ldots, \varphi_k(k)$ terminates in at most n steps,

(ii) $\varphi_k(0) = a_0$, $\varphi_k(1) = a_1, \ldots, \varphi_k(k) = a_k$,

(iii) $(\exists l)((k \le l \le n)\ \&\ (a_{i-1} \ne a_i))$

If such a k exists, then the function $a_0a_1\ldots a_n a_n^\infty$ is removed from $\mathcal{U}'$. If such a k does not exist, the function $a_0a_1\ldots a_n a_n^\infty$ remains in $\mathcal{U}'$, and we place the following pairs into T:

$$\begin{array}{l} \langle\{a_0a_1\ldots a_n a_n\}, n+1\rangle \\ \langle\{a_0a_1\ldots a_n(a_n+1)\}, n+1\rangle \\ \langle\{a_0a_1\ldots a_n(a_n+2)\}, n+1\rangle \\ \cdots \\ \langle\{a_0a_1\ldots a_n(a_n+n+3)\}, n+1\rangle, \end{array}$$

and we additionally place the following $n+3$ functions into $\mathcal{U}'$:

$$\begin{array}{l} a_0a_1\ldots a_n(a_n+1)^\infty \\ a_0a_1\ldots a_n(a_n+2)^\infty \\ \cdots \\ a_0a_1\ldots a_n(a_n+n+3)^\infty \end{array}$$

Go to Stage $(n+2)$.
End Stage $n+1$.

Now we prove that the class $\mathcal{U}$ is nonempty. Indeed, the functions 0^∞ and 1^∞ cannot be removed from $\mathcal{U}'$ because of (iii). We define the desired effective operation by defining a partial recursive function $\psi(x)$. To compute this value, in parallel generate the pairs in T, and compute $\varphi_x(0), \varphi_x(1), \ldots$. Let the pair

$$\langle\{a_0a_1\ldots a_n\}, b\rangle \in T$$

Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

be the first pair (according to our parallel computation) such that $\varphi_x^{[n]} = a_0a_1 \dots a_n$ and $x \leq b$. Let m be the largest integer $m \leq n$ such that $a_m = a_{m+1} = \dots = a_n$. Then we define $\psi(x) = s(a_0a_1 \cdots a_m)$.

Now we prove that ψ defines an effective operation on $\mathcal{U}$. (Please notice that ψ does not define an effective operation on $\mathcal{R}$.) Let $\varphi_x \in \mathcal{U}$. If φ_x is a constant function $\varphi_x(t) \doteq a$ then, by our definition, $\psi(x) = s(a)$, i. e. $\psi(x)$ does not depend on the particular φ-index of the function. If φ_x is a non-constant function, then $\varphi_x = a_0a_1 \dots a_n a_{n+1}^{\infty}$. Assume that in this notation n is chosen such that $a_n \neq a_{n+1}$. In this case $x \geq n+1$ (otherwise the function would have been removed from $\mathcal{U}'$). However for all $b \geq n+1$ the pair $\langle \{a_0a_1 \dots a_b\}, b \rangle$ compatible with the function φ_x leads to representation

$$a_0a_1 \dots a_b = a_0a_1 \dots a_n a_{n+1} \dots a_b$$

with $a_{n+1} = a_{n+2} = \dots = a_b$. This implies $\psi(x) = s(a_0a_1 \dots a_n)$.

Now we prove the property b) of the class $\mathcal{U}$. Let the function $f = a_0a_1 \dots a_n a_{n+1}^{\infty}$ be in $\mathcal{U}$ $(a_n \neq a_{n+1})$, and let $f^{[x]}$ be the initial segment from the property b). If $x \geq n+1$, then at Stage x the functions

$$\begin{array}{l} a_0a_1 \dots a_n a_{n+1} \dots a_{x-1}(a_x+1)^{\infty} \\ a_0a_1 \dots a_n a_{n+1} \dots a_{x-1}(a_x+2)^{\infty} \\ \dots \\ a_0a_1 \dots a_n a_{n+1} \dots a_{x-1}(a_x+x+2)^{\infty} \end{array}$$

are placed into $\mathcal{U}'$.

At least one of these $x+2$ functions is such that its minimum φ-index exceeds $x+1$. This function g is different from f, and it is never removed from $\mathcal{U}'$. Hence this function is in $\mathcal{U}$, and $g^{[x]} = f^{[x]}$.

If $x < n+1$, then consider the function $f = a_0a_1 \dots a_n a_{n+1}^{\infty} \in \mathcal{U}$ and the functions

$$\begin{array}{l} a_0a_1 \dots a_x(a_x+1)^{\infty} \\ a_0a_1 \dots a_x(a_x+2)^{\infty} \\ \dots \\ a_0a_1 \dots a_x(a_x+x+3)^{\infty} \end{array}$$

placed into $\mathcal{U}'$ at Stage $x+1$. At least one of these $x+3$ functions is such that its minimum φ-index is at least $x+1$. Hence this function g is never removed from $\mathcal{U}'$, and $g^{[x]} = f^{[x]}$. □

Theorem 11 *If a class $\mathcal{U}$ of total recursive functions is in FIN^+, then $\mathcal{U}$ is in BC.*

4 Incompatibility of the two generalization lines

The proof of the subsequent theorem is based on the mutual recursion theorem of R. Smullyan [Smu 61], see also [Smi 94].

Theorem 12 *There is a class $\mathcal{U}$ of total recursive functions such that :*

1. *$\mathcal{U} \in randFIN$,*
2. *$\mathcal{U} \notin FIN^+$.*

The identification types $randFIN$ and FIN^+ are independent. On one hand it follows from Theorem 12. On the other hand, we had Theorem 8 saying that $FIN^+ \setminus randFIN$ is not empty.

Now we have come to the result we consider as the most unexpected. The two gerneralization lines (one represented by FIN and the other one represented by $FSTAND$) turn out to be incomparable.

Theorem 13 *If a class $\mathcal{U}$ of total recursive functions is both in FIN^+ and in $randFIN$, then $\mathcal{U}$ is in FIN.*

Proof. To identify $\mathcal{U}$ by a deterministic identification machine in FIN-mode, simulate the work of the $randFIN$-machine until the total probability strictly exceeds $\frac{1}{2}$. Take the maximum of all the obtained results and use the FIN^+-machine with this upper bound of a correct program. The upper bound is surely correct by the definition of the $randFIN$-identification (the total probability of the correct programs exceeds $\frac{1}{2}$). Hence the result by the FIN^+-machine is also to be correct. □

References

[AS 83] Dana Angluin, and Carl H. Smith. Inductive inference: theory and methods. *Computing Surveys,* vol. 15, 1983, pp. 237–269.

[Bar 74] J. Barzdins, Two Theorems on the Limiting Synthesis of Functions, In *Theory of Algorithms and Programs,* edited by J. Barzdins, Latvian State University, 1974, pp. 82-88.

[Fre 78] Rūsiņš Freivalds. Effective operations and functionals computable in the limit. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik,* vol. 24, 1978, pp. 193–206 (in Russian)

[Fre 79] Rūsiņš Freivalds. Finite identification of general recursive functions by probabilistic strategies. *Proceedings of FCT'79. Mathematical Research. Akademie-Verlag, Berlin,* vol. 2, pp. 138–145, 1979

[FKW 84] Rūsiņš Freivalds, Efim B. Kinber. and Rolf Wiehagen. Connections between identifying functionals, standardizing operations and computable numberings. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik,* vol. 30, 1984, pp. 145–164.

[FKW 89] Rūsiņš Freivalds, Efim B. Kinber, and Rolf Wiehagen. Inductive inference from good examples. *Lecture Notes in Artificial Intelligence*, vol. 397, pp. 1–17, 1989.

[FKW 93] Rūsiņš Freivalds, Efim B. Kinber, and Rolf Wiehagen. On the power of inductive inference from good examples. *Theoretical Computer Science*, vol. 110, pp. 131–144, 1993

[FKS 94] Rūsiņš Freivalds, Marek Karpinski, and Carl H. Smith. Co-learning of total recursive functions. *Proceedings of the Seventh Annual Conference on Computational Learning Theory*, New Brunswick, New Jersey, pages 190-197. ACM-Press, July 1994.

[FW 79] Rūsiņš Freivalds, and Rolf Wiehagen, Inductive inference with additional information. *Journal of Information Processing and Cybernetics* , vol. 15, pp. 179–185, 1979

[FWB 94] Rūsiņš Freivalds, Ognian Botuscharov, and Rolf Wiehagen. Identifying nearly minimal Gödel numbers from additional information. *Lecture Notes in Computer Science*, vol. 872, pp. 91–99, 1994

[FS 93] Rūsiņš Freivalds and Carl H. Smith. On the role of procrastination in machine learning. *Information and Computation*, vol. 107, pp. 237–271, 1993

[FKS 95] Rūsiņš Freivalds and Efim Kinber and Carl H. Smith. On the impact of forgetting on learning machines. *Journal of the ACM*, vol. 42, pp. 1146–1168, 1995

[Go 67] E. M. Gold. Language identification in the limit. *Information and Control*, vol. 10, pp. 447–474, 1967

[KLS 57] Georg Kreisel, Daniel Lacombe, and Joseph R. Shoenfield. Partial recursive functionals and effective operations. *Constructivity in mathematics: proceedings of the colloquium held at Amsterdam, A.Heyting (ed.) North-Holland*, pp. 195–207, 1957

[LZ 92] Stefan Lange and Thomas Zeugmann. Types of monotonic language learning and their characterization. *Proceedings of COLT'92*, pp. 377–390, 1992

[LZ 93] Stefan Lange and Thomas Zeugmann. Learning recursive languages with a bounded number of mind changes. *International Journal of Foundations of Computer Science*, vol. 4, pp. 157–178, 1993

[LZ 93a] Stefan Lange and Thomas Zeugmann. Language learning with a bounded number of mind changes. *Lecture Notes in Computer Science*, vol. 665, pp. 682–691, 1993

[Meh 73] Kurt Mehlhorn. On the size of sets of computable functions. *Proceedings of the 14th Symposium on Switching and Automata Theory*, pp. 190–196, IEEE Computer Society, 1973

[OSW 86] Daniel N. Osherson, Michael Stob, and Scott Weinstein. Systems that Learn. MIT Press, 1986.

[Ro 67] Hartley Rogers Jr. Theory of Recursive Functions and Effective Computability. McGraw-Hill, 1967.

[Smi 94] Carl H. Smith. *A Recursive Introduction to the Theory of Computation.* Springer-Verlag, 1994

[Smu 61] R. Smullyan, *Theory of Formal Systems, Annals of Mathematical Studies*, Princeton University Press, Vol. 47, 1961.

[Wie 75] Rolf Wiehagen. Inductive inference of recursive functions. *Lecture Notes in Computer Science*, vol. 32, pp. 462–464, 1975

[Wie 78] Rolf Wiehagen. Zur Theorie der Algorithmischen Erkennung. *Ph.D. Thesis, Humboldt Universität, Berlin*, 1978

[WFK 84] Rolf Wiehagen, Rūsiņš Freivalds and Efim B. Kinber. On the power of probabilistic strategies in inductive inference. *Theoretical Computer Science*, vol. 25, pp. 111–133, 1984

[Zeu 86] Thomas Zeugmann. On Barzdin's conjecture. *Lecture Notes in Computer Science*, vol. 265, pp. 117–125, 1986

[Zeu 91] Thomas Zeugmann. Inductive inference of optimal programs. *Lecture Notes in Computer Science*, vol. 543, pp.208–222, 1991

[ZL 95] Thomas Zeugmann and Stefan Lange. A guided tour across the boundaries of learning recursive languages. *Lecture Notes in Computer Science*, vol. 961, pp. 193–262, 1995