Inductive Inference of Limiting Programs with Bounded Number of Mind Changes *

Juris Vīksna
Institute of Mathematics and Computer Science
The University of Latvia
Rainis boulevard 29
Riga, LV - 1459
LATVIA
email jviksna@cclu.lv

Abstract

We consider inductive inference of total recursive functions in the case, when produced hypotheses are allowed some finite number of times to change "their mind" about each value of identifiable function. Such type of identification, which we call *inductive inference of limiting programs with bounded number of mind changes*, by its power lies somewhere between the traditional criteria of inductive inference and recently introduced inference of limiting programs.

We consider such model of inductive inference for EX and BC types of identification, and we study

- tradeoffs between the number of allowed mind changes and the number of anomalies, and
- relations between classes of functions identifiable with different probabilities.

For the case of probabilistic identification we establish probabilistic hierarchies which are quite unusual for EX and BC types of inference.

1 Introduction

In many real life situations, when a human being is asked about something, she at first can give a wrong answer to the question and only later (after some thinking) she can change it to a correct one. The number of wrong answers that appear before the correct one can be even larger. Nevertheless, if somebody is able to give a correct answer, though only in the way described above, we usually will agree that the person "has learned" the thing she was asked about.

In this paper we investigate a model of inductive inference that attempts to formalize the learning process informally discussed above.

We consider algorithmic identification of limiting programs of recursive functions. By limiting program for a function f we understand a program that, when computing the value of $f(x)$, on any input x is allowed to change "its mind" a finite number of times. As identification devices we use inductive inference machines (IIM) that as hypotheses produce Gödel numbers of recursive functions with two variables. Identification by such inductive inference machine is considered to be successful, if for a given function f IIM has produced a natural number h, such that for each argument value x there exists a natural number $n(x)$, with $0 \leq n(x) < \nu$

*Part of this work was done during the author's stay at University of Delaware and was supported by the Department of Computer and Information Sciences.

(where ν describes the number of allowed mind changes and is either natural positive, or "infinite", i.e. in our notation $\nu \in \mathbf{N}_+ \cup \{\omega\}$), such that:

1. $\varphi_h(x, n(x)) = f(x)$,
2. $\forall k \in \mathbf{N}_+, k < n(x) : \varphi_h(x, k)$ converges, and
3. $\forall k \in \mathbf{N}_+, n(x) < k < \nu : \varphi_h(x, k)$ diverges.

It is clear that the case $\nu = 1$ corresponds to traditional criteria of identification of exact (non-limiting) programs, such as FIN, EX or BC, because already the first guess about each value $f(x)$ is required to be correct. The case $\nu = \omega$ (by which we understand that we do not place upper bound on $n(x)$) corresponds to identification of limiting programs, introduced in [2] and formalizes the situation where for each question about $f(x)$ we can change our answer any finite number of times and only the last answer must be correct. If ν is finite and $\nu > 1$, then we obtain types of identification that by their power lie somewhere between those mentioned above. In this situation for each value $f(x)$ we can produce up to ν different answers and only the last one has to be correct. This kind of identification here is referred to as *inductive inference of limiting programs with bounded number of mind changes.* The name can be somewhat misleading, because traditionally by inductive inference with bounded number of mind changes we understand EX identification, when a number of allowed hypotheses for identifiable function f (not for each value $f(x)$ of this function) is bounded by some natural number. However, it seems that it is quite hard to find another sufficiently adequate and short name, therefore here we shall use the notion "mind changes" in the above described meaning.

In this paper we shall study inference of limiting programs for EX and BC types of identification, i.e.

- in the case, when for successful identification of a given function IIM is allowed to produce a finite number of hypotheses and the last of them has to be a correct one (EX identification), and
- in the case, when for successful identification of a given function IIM is allowed to produce an infinite number of hypotheses and all but finite number of them have to be correct (BC identification).

It might appear that inductive inference of limiting programs (at least for types EX and BC) is too general to describe any real learning processes, so that we obtain only in the limit some program that, again only in the limit, computes each value of identifiable function. However, if, for example, we can EX identify a limiting number of some function, then after a finite time of computation we have an algorithm that at least somehow can be used to compute this function. At the same time, in the case of BC identification of exact (non-limiting) numbers we can be forced to change an algorithm for computing our function infinitely many times. Thus, in some sense EX identification of limiting numbers gives us more information about identifiable function than BC identification of exact numbers.

For identification of limiting programs in probabilistic case we obtain some quite unusual probabilistic hierarchies for EX and BC types of identification. In general, relations between classes of functions, identifiable with different probabilities, strongly depend on the particular type of identification. However, it turns out that in some situations we obtain the same probabilistic hierarchies for EX identification of limiting programs as we have for FIN identification of exact programs. Thus, it seems that such results in some sense can help us better understand the effects of probabilistic identification in learning processes.

For BC type of identification inductive inference of limiting programs with bounded number of mind changes also appears to be a very natural generalization, because BC identification of limiting programs (without upper bound on the number of mind changes) turns out to be too broad, so that it is known that in this way the class $\mathcal{R}$ of all total recursive functions is identifiable.

2 Notation and definitions

Our notation and terminology are standard and, in general, follow [7], with respect to the theory of recursive functions and [3], [8], etc., with respect to the theory of inductive inference.

We consider some fixed Gödel numbering φ of all partial recursive functions. For technical reasons we additionally shall assume that function $\varphi_0(x)$ is undefined for any argument value x.

For the task of learning we use inductive inference machines (IIM) – a special type of Turing machines with input tape, working tape and output tape, which receive a graph of some recursive function f on the input tape, and which can, from time to time, print some number on the output tape. These numbers we shall consider as current hypotheses about the identifiable function f.

By $\mathcal{R}$ and $\mathcal{P}$ we denote, correspondingly, the class of all total recursive functions and the class of all partial recursive functions with one variable. Similarly, by $\mathcal{R}_n$ and $\mathcal{P}_n$ we denote analogous classes of functions with n variables.

By $\mathbf{N}$ and $\mathbf{N}_+$ we denote the set of all natural numbers and the set of all positive natural numbers. The first infinite ordinal we denote by ω (here we shall understand it only as a "number" that is larger than any natural number, i.e. for all $x \in \mathbf{N}$ we have $x < \omega$).

$< \cdot, \ldots, \cdot >$: $\mathbf{N}^n \rightarrow \mathbf{N}$, where $n \in \mathbf{N}_+$ will be some fixed computable one to one mappings between the set of all$n$- tuples of natural numbers $\mathbf{N}$ and the set of all natural numbers $\mathbf{N}^n$, such that the inverse mappings of $< \cdot, \ldots, \cdot >$ are also computable. For each natural number x we denote by $x[1], \ldots, x[n]$ natural numbers, such that $x =< x[1], \ldots, x[n] >$. (It is clear that $x[i]$ will be well defined only if the value of n is previously fixed. Thus, when we use this notation we shall always explicitly mention which value of n we are using.)

By $f(x_1, \ldots, x_n) \downarrow$ and $f(x_1, \ldots, x_n) \uparrow$ we understand that function $f \in \mathcal{P}_n$, correspondingly, converges or diverges on value $(x_1, \ldots, x_n) \in \mathbf{N}^n$.

The difference of two sets A and B we denote by $A - B$, i.e., if A and B are sets, then $A - B$ is a set $\{x \mid x \in A, x \notin B\}$. By $A \subset B$ we understand that A is a subset (probably not proper) of B.

By $f[k]$, where $f \in \mathcal{R}$ and $k \in \mathbf{N}$, we denote the initial sequence of $k+1$ elements from the graph of function f, i.e. $f(0), f(1), \ldots, f(k)$. The hypothesis produced by some inductive inference machine M after the processing of the sequence $f[k]$ we denote by $M(f[k])$. In general $M(f[k])$ can also be undefined, in this case we write $M(f[k]) \uparrow$.

For all $f \in \mathcal{P}_2$ and for all $\nu \in \mathbf{N}_+ \cup \{\omega\}$ we define function f^ν in the following way:

$$\forall x \in \mathbf{N}: \ f^\nu(x) = \begin{cases} f(x, s-1), & \text{where } s = max(\{t \in \mathbf{N}, 0 \leq t < \nu \mid f(x,t) \downarrow\}), \\ & \text{if } s \text{ is defined and } s \in \mathbf{N}, \\ \uparrow, & \text{otherwise.} \end{cases}$$

Instead of f^ω we usually write $f^\star$.

Thus, equality $g = f^k$, where $k \in \mathbf{N}$ and $f \in \mathcal{P}_2$, could be interpreted as the fact that f computes function g with at most $k-1$ mind changes. Similarly, equality $g = f^\omega$ could be interpreted as the fact that f computes function g with a finite number of mind changes.

By $f^{\nu,m}(x)$ we denote the m-th guess computed by f^ν, i.e. the value $f(x, m-1)$.

For all $k \in \mathbf{N}$ we call two functions $f, g \in \mathcal{P}$ to be equal up to a set with k elements, if

$$card(\{x \in \mathbf{N} \mid f(x) \neq g(x) \ \vee \ f(x) \uparrow\}) \leq k.$$

Such an equality we denote by $f =^k g$.

Similarly, we call two functions $f, g \in \mathcal{P}$ to be equal up to a finite set if

$$card(\{x \in \mathbf{N} \mid f(x) \neq g(x) \ \vee \ f(x) \uparrow\}) < \omega.$$

Such an equality we denote by $f =^\omega g$ or $f =^\star g$.

For an arbitrary recursive function with n arguments $f \in \mathcal{P}_n$ by $\#f$ we shall understand a natural number, such that $\varphi_{\#f} = f$.

In proofs of our theorems often we shall use the recursion theorem that allows in situations when we are defining a finite number of recursive functions $f_1, \ldots, f_n$ by describing some algorithm, which computes these functions, to use pairs $(x, \#f_n)$ as inputs for this algorithm to compute the values $f_n(x)$.

For brevity in such cases usually we shall not mention explicitly the use of the recursion theorem and simply use the values $\#f_1, \ldots, \#f_n$ as arguments of our algorithm for defining functions $f_1, \ldots, f_n$. Naturally, in all these cases we are able to rewrite our proofs in more formal way.

Besides that, we quite often shall need to simulate k steps of computation of some value $f(x)$ of partial recursive function f. For this reason we define a notation of function $f\langle k\rangle$ in the following way.

Let $n \in \mathbf{N}_+$ and let $f \in \mathcal{P}_n$. Then

$$\forall x \in \mathbf{N}^n, \forall k \in \mathbf{N}_+ :\ f\langle k\rangle(x) = \begin{cases} f(x), & \text{if after } k \text{ steps of computation } f(x) \\ & \text{converges,} \\ \uparrow, & \text{otherwise.} \end{cases}$$

In our definitions of EX types of identification, which are given below, we require an IIM to produce an infinite sequence of hypotheses (i.e. for function f, given for identification to inductive inference machine M, values $M(f[x])$ are required to be defined for all $x \in \mathbf{N}$) all but finite number of which must be equal to some fixed natural number. It is done for technical reasons, and it is easy to see that such definition of EX identification is identical to the traditional one when IIM is allowed to produce only finite sequence of hypotheses. BC types of identification will be defined in the usual way.

Definition 1 *A subset A of total recursive functions is $EX[\nu]$- identifiable (where $\nu \in \mathbf{N}_+ \cup \{\omega\}$), if there exists an inductive inference machine M, such that for every function $f \in A$ the machine M, when given graph of f, outputs an infinite sequence of natural numbers $i_0, i_1, \ldots, i_k, \ldots$, such that there exists $k_0 \in \mathbf{N}$, such that for all $k \geq k_0$ we have $i_k = i_{k_0}$ and $f = \varphi^{\nu}_{i_k}$.* $\square$

We denote the class of all sets of functions identifiable in this way by $EX[\nu]$.

Definition 2 *A subset A of total recursive functions is $BC[\nu]$- identifiable (where $\nu \in \mathbf{N}_+ \cup \{\omega\}$), if there exists an inductive inference machine M, such that for every function $f \in A$, the machine M, when given graph of f, outputs an infinite sequence of natural numbers $i_1, i_2, \ldots, i_k, \ldots$, such that there exists $k_0 \in \mathbf{N}$, such that for all $k \geq k_0$ we have $f = \varphi^{\nu}_{i_k}$.* $\square$

We denote the class of all sets of functions identifiable in this way by $BC[\nu]$.

We also consider inference of limiting programs with anomalies, i.e. the case when IIM is allowed to produce an answer that can be wrong for some finite number of values from the domain of the identifiable function.

Definition 3 *A subset A of total recursive functions is $EX^a[\nu]$- identifiable (where $\nu \in \mathbf{N}_+ \cup \{\omega\}$ and $a \in \mathbf{N} \cup \{\omega\}$), if there exists an inductive inference machine M, such that for every function $f \in A$ the machine M, when given graph of f, outputs an infinite sequence of natural numbers $i_0, i_1, \ldots, i_k, \ldots$, such that there exists $k_0 \in \mathbf{N}$, such that for all $k \geq k_0$ we have $i_k = i_{k_0}$ and $f =^a \varphi^{\nu}_{i_k}$.* $\square$

We denote the class of all sets of functions identifiable in this way by $EX^a[\nu]$.

In considering probabilistic identification we assume that IIM is a probabilistic Turing machine that in each step can act in several different ways and with each of these ways there is associated a real number from the interval $[0,1]$ (which describes the probability with which this way is chosen), such that the sum of numbers corresponding to all possible actions is equal to 1.

Thus, the probability p that inductive inference machine will output a sequence $i_0, i_1, \ldots, i_k, \ldots$, when working on the graph $f(0), f(1), \ldots, f(k), \ldots$ of function f will be equal with

$$\left[\begin{array}{c}\text{probability that IIM}\\ \text{will output } i_0 \text{ on } f[0]\end{array}\right] \cdot \prod_{k\in\mathbf{N}} \left[\begin{array}{c}\text{probability that IIM will output}\\ i_{k+1} \text{ on } f[k+1] \text{ if until so far it}\\ \text{has produced a sequence } i_0,\ldots,i_k.\end{array}\right]$$

As far as we know, in all cases described in this paper probabilistic inference can be substituted also by team inference, i.e. for any set of functions A, identifiable by some IIM with probability $p = r/s$, there also exists a team T of s deterministic inductive inference machines, such that each function $f \in A$ can be correctly identified by at least r inductive inference machines from team T. At the same time, that does not necessarily imply that such simplification is possible in all situations, i.e. also in cases that are not covered by our results.

Definition 4 *A subset A of total recursive functions is $EX[\nu]$- identifiable (where $\nu \in \mathbf{N}_+ \cup \{\omega\}$) with probability $p \in [0,1]$, if there exists an inductive inference machine M, such that for every function $f \in A$ the machine M, when given graph of f, with probability p outputs an infinite sequence of natural numbers $i_0, i_1, \ldots, i_k, \ldots$, such that there exists $k_0 \in \mathbf{N}$, such that for all $k \geq k_0$ we have $i_k = i_{k_0}$ and $f = \varphi^{\nu}_{i_k}$.* $\square$

We denote the class of all sets of functions identifiable in this way by $EX_p[\nu]$.

As we have already mentioned earlier and as it easily follows from definitions, we have the identities $EX[1] = EX$, $BC[1] = BC$, and $EX[\omega] = LimEX$, $BC[\omega] = LimBC$, where $LimEX$ and $LimBC$ are classes of inference of limiting programs introduced in [2]. Also, to follow the tradition, we write $EX[\star]$ instead of $EX[\omega]$, $BC[\star]$ instead of $BC[\omega]$ and $EX^{\star}[\nu]$ instead of $EX^{\omega}[\nu]$.

Definitions of the classes $BC^a[\nu]$ and $BC_p[\nu]$ are similar to the definitions of the classes $EX^a[\nu]$ and $EX_p[\nu]$. Further we shall consider classes $BC^a[\nu]$ and $BC_p[\nu]$ only with parameters $a \in \mathbf{N}$ and $\nu \in \mathbf{N}_+$, because it is known that classes $BC^{\star}$ and $BC[\star]$ each contains the set $\mathcal{R}$ of all total recursive functions. Hence the classes $BC^{\star}[\nu]$, $BC^a[\star]$ and $BC_p[\star]$ are trivial and not interesting.

In the following sections quite often we shall give the proofs only for some special cases of our theorems. It will be done in cases where such approach will result in significantly shorter and more readable proofs, and at the same time proofs of the general cases will differ only in technical complexity, but will not require any new ideas, and we expect that they quite easily could be restored by the reader.

3 Inductive inference of limiting programs with anomalies

In this section we shall study relations between classes of functions, identifiable with different numbers of mind changes and different number of anomalies, in deterministic case. In general, we might expect that, if we are allowed to produce programs that can make mind changes about each value of identifiable program, then probably we can trade number of mind changes to reduce number of anomalies. However, it turns out not to be the case. In [2] it is already shown that

1. $BC - EX^{\star}[\star] \neq \emptyset$;
2. $EX^{\star}[\star] - BC \neq \emptyset$;

3. $\mathcal{R} \in BC[\star]$;

4. $\forall a \in \mathbf{N} : \; EX^{a+1} - EX^{a}[\star] \neq \emptyset$.

Here we additionally prove the following results, which almost complete the picture for the generalized case. The first theorem shows that by increasing the number of allowed mind changes we increase the learning power. This increase cannot be compensated even by using BC identification instead of EX.

Theorem 1 *For all $n \in \mathbf{N}_+ : \; EX[n+1] - BC[n] \neq \emptyset$.* $\Box$

Proof. We define the set A_n, such that $A_n \in EX[n+1]$ and $A_n \notin BC[n]$, in the following way:

$$A_n = \{f \in \mathcal{R} \mid f = \varphi^{n+1}_{f(0)}\}.$$

For brevity we shall show here that the set A_n have desired properties only for the case when $n = 2$.

Let $A = A_2$.

It is clear that $A \in EX[3]$, so that each function $f \in A$ is $EX[3]$ identifiable by inductive inference machine, which as hypothesis about function f outputs the sequence $f(0), f(0), \ldots$.

Now we shall show that A does not belong to the class $BC[2]$. For each inductive inference machine M we shall construct a function $f_M \in A$, such that machine M does not $BC[2]$ identify function f_M.

We define three recursive functions f_0, f_1, f_2 by the following procedure.

Step 1.
We define $H_1 = \emptyset$, $H_2 = \emptyset$ and for all $t \in \{0,1,2\}$ define $f_t(0) = \#\xi$, where function $\xi \in \mathcal{P}_2$ is defined in the following way:

$$\forall x, t \in \mathbf{N} : \; \xi(x,t) = \begin{cases} f_t(x), & \text{if } 0 \leq t \leq 2, \\ \uparrow, & \text{otherwise.} \end{cases}$$

(Here sets H_t informally could be described as the sets containing the hypotheses of machine M on functions which we are defining together with the information about the initial segments on which these hypotheses are produced.)

Step $k, k > 1$.
We define $f_0(k) = 0$. Then we compute the value $h(k) = M(F_k[k-1])$, i.e. the hypothesis produced by M on the first k values from the graph of function F_k, where function F_k is defined in the following way:

$$\forall x \in \mathbf{N} : \; F_k(x) = \begin{cases} f_{2,k}(x), & \text{if } f_{2,k}(x) \downarrow, \\ f_{1,k}(x), & \text{if } f_{2,k}(x) \uparrow \text{ and } f_{1,k}(x) \downarrow, \\ f_{0,k}(x), & \text{otherwise,} \end{cases}$$

where functions $f_{t,k}$ correspond to the parts of the functions f_t that are already defined until the k- th step. Then we define $H_1 = H \cup \{(h(k), k)\}$ and $H_2 = H \cup \{(h(k), k)\}$.

Further for each $t \in \{1,2\}$ and for each pair $(h, k_h) \in H_t$ we compute the value $\varphi_h\langle k\rangle(k_h, t)$. If there exists a pair $(h_t, k_{h_t}) \in H_t$, such that $\varphi_{h_t}\langle k\rangle(k_{h_t}, t) \downarrow$, then we define $f_t(k_{h_t}) = \varphi_{h_t}\langle k\rangle(k_{h_t}, t) + 1$ and for $s \in \{1,2\}$ define $H_s = \{(h, k_h) \in H_s \mid k_h < k_{h_t}\}$.

Further, sequentially for $t = 1$ and $t = 2$, if there exists $y \in \mathbf{N}$, such that $f_t(y)$ is already defined and $f_t(z)$ is undefined for some $z < y$, then for all $x \leq y$ we define

$$f_t(x) = \begin{cases} f_{t,k}(x), & \text{if } f_{t,k}(x) \downarrow, \\ f_{s,k}(x), & \text{if } 0 < s < t, f_{s,k}(x) \downarrow \text{ and } f_{t,k}(x) \uparrow, \\ f_{0,k}(x), & \text{otherwise.} \end{cases}$$

Then we go to the next step.
End.

Now we consider the function f_M defined by

$$\forall x \in \mathbf{N}: \ f_M(x) = \begin{cases} f_2(x), & \text{if } f_2(x) \downarrow, \\ f_1(x), & \text{if } f_2(x) \uparrow \text{ and } f_1(x) \downarrow, \\ f_0(x), & \text{otherwise.} \end{cases}$$

If there exists $t \in \{1,2\}$, such that function f_t is total, then for some $s \in \{1,2\}$ we have $f_M(x) = f_s(x)$ for all but finite number of values $x \in \mathbf{N}$, hence f_M is recursive. If there does not exist such $t \in \{1,2\}$, then $f_M(x)$ is equal to $f_0(x)$ for all but finite number of values $x \in \mathbf{N}$, hence also in this case f_M is recursive. From construction of f_M it follows that machine M does not $BC[2]$ identify function f_M. Besides that, by definition of $f_M(0)$ we have that $f_M = \varphi^3_{f_M(0)}$. Therefore, function $f_M \in A$, while it is not $BC[2]$ identifiable by machine M.□

Corollary 1 *For all* $a \in \mathbf{N}, n \in \mathbf{N}_+ : \ EX[n+1] - BC^a[n] \neq \emptyset$. □

Proof. Let $A \in EX[n+1] - BC[n]$. By the previous theorem such set A exists. We consider set

$$A' = \{g \mid \exists f \in A : \ [\forall x, y \in \mathbf{N} : \ g(<x,y>) = f(x)]\}.$$

Clearly $A' \in EX[n+1]$. Besides that, if $A' \in BC^a[n]$, then $A \in BC[n]$, that contradicts the choice of A. Therefore $A' \notin BC^a[n]$.□

From the results given above we can also conclude that, if $a \in \mathbf{N}, n \in \mathbf{N}_+$, then $\mathcal{R} \notin BC^a[n]$, i.e. classes $BC^a[n]$ are not trivial (in the contrary to $\mathcal{R} \in BC[\star]$ shown in [2]).

The next theorem is a simple generalization of already known result that $EX^\star \subset BC$, i.e. any finite number of errors for $EX[n]$ identification can be compensated by using $BC[n]$ identification.

Theorem 2 *For all* $n \in \mathbf{N}_+ : \ EX^\star[n] \subset BC[n]$, *where inclusion is proper.* □

In conclusion we can establish that classes $EX^a[n]$ and $BC^a[n]$ form a strongly increasing hierarchies, as increases the value of n.

Corollary 2 *For all* $a \in \mathbf{N} \cup \{\star\} : \ EX^a[1] \subset EX^a[2] \subset \cdots \subset EX^a[\star]$, *where all inclusions are proper.* □

Corollary 3 *For all* $a \in \mathbf{N} : \ BC^a[1] \subset BC^a[2] \subset \cdots \subset BC^a[\star]$, *where all inclusions are proper, and* $\mathcal{R} \in BC^a[\star]$. □

In contrary to the already mentioned result from [2] which holds for $EX[\nu]$ identification, namely $\forall a \in \mathbf{N} : \ EX^{a+1} - EX^a[\star] \neq \emptyset$, the situation for $BC[\nu]$ identification with errors is more complicated.

Theorem 3 *For all* $n \in \mathbf{N}$ *the following holds:*

1. $BC^n \subset BC[n+1]$, *and*
2. $BC^n - BC[n] \neq \emptyset$.□

Proof. (1) Let $A \in BC^n$ and let M be IIM that BC^n identifies A. We shall construct another inductive inference machine M', such that M' $BC[n+1]$ identifies set A.

Let for each $f \in A$ and for each $k \in \mathbf{N}$ sets $E_{f,k,0}, \ldots, E_{f,k,n}$ to be defined by equalities

$$E_{f,k,i} = \{M(f[t]) \mid t \leq k \ \&\ card(\{s \leq k \mid \varphi_{M(f[t])}\langle k\rangle(s) \downarrow\neq f(s)\}) = i\}.$$

Let $last(E_{f,k,i}) = \{M(f[t]) \mid M(f[t]) \in E_{f,k,i} \ \&\ \forall t < s \leq k :\ M(f[s]) \notin E_{f,k,i}\}$, i.e. $last(E_{f,k,i})$ will be the last hypothesis placed in $E_{f,k,i}$.

Inductive inference machine M' for each $k \in \mathbf{N}$ as hypothesis $M(f[k])$ will output number of function $\xi \in \mathcal{P}_2$ defined by induction (by k) as follows:

$$\forall x \in \mathbf{N}, \forall k \leq n :\ \xi(x,k) = \begin{cases} f(x), & \text{if } x \leq k, \\ last(E_{f,x,k}), & \text{if } E_{f,x,k} \neq \emptyset, \\ \xi(x,k+1), & \text{if } E_{f,x,k} = \emptyset \ \&\ k < n, \\ \uparrow, & \text{otherwise.} \end{cases}$$

Let $a \in \{0,\ldots,n\}$ be the largest number, such that for some $k \in \mathbf{N}$ we have

$$card(\{t \in \mathbf{N} \mid \varphi_{M(f[k])}(t) \downarrow\neq f(t)\}) = a$$

and $f =^n \varphi_{M(f[t])}$ for all $t \geq k$.

Then for all sufficiently large x we shall have equality $\varphi_{last(E_{f,x,a})}(x) = f(x)$. At the same time, for all i with $a < i \leq n$ for all sufficiently large x we shall have that either $\varphi_{last(E_{f,x,i})}(x) \uparrow$ or $\varphi_{last(E_{f,x,i})}(x) = f(x)$.

Thus for all sufficiently large x also the equality $f = \varphi^{a+1}_{last(E_{f,x,a})}$ will hold, i.e. M' will $BC[n+1]$ identify function f. Therefore $A \in BC[n+1]$.

(2) The proof of this case can be obtained as generalization of proof for already known result that $BC^1 - BC[1] \neq \emptyset$ (see [3]). We shall show here proof only for the case $n = 2$.

Let the set A be defined in the following way:

$$A = \{f \in \mathcal{R} \mid \exists x_0 \in \mathbf{N} :\ f(x_0)[1] \neq 0 \ \&\ \forall x \geq x_0 :\ [f(x)[1] \neq 0 \Rightarrow f =^2 \varphi_{f(x)[1]}]\}.$$

It is easy to see that $A \in BC^2[1]$, because each function $f \in A$ can be $BC^2[1]$ identified by inductive inference machine M, which for all $x \in \mathbf{N}$ outputs hypothesis $M(f[x])$ defined as

$$M(f[x]) = \begin{cases} f(x)[1], & \text{if } f(x)[1] \neq 0, \\ M(f[x-1]), & \text{if } f(x)[1] = 0 \text{ and } x > 0, \\ 0, & \text{otherwise.} \end{cases}$$

We shall show that $A \notin BC[2]$. For each inductive inference M machine we shall construct a function f_M, such that machine M does not $BC[2]$ identify function f_M.

For an arbitrary function $f \in \mathcal{P}_m$, for some natural number m, by $f\{(x_1,y_1),\ldots,(x_k,y_k)\}$ (where $x_i \in \mathbf{N}^m$ and $y_i \in \mathbf{N}$) we shall denote the function defined by

$$f\{(x_1,y_1),\ldots,(x_k,y_k)\}(x) = \begin{cases} y_i, & \text{if } x = x_i \text{ for some } i, \text{ with } 1 \leq i \leq k, \\ f(x), & \text{otherwise.} \end{cases}$$

We define three functions f_0, f_1 and f_2 together with function F by the following procedure.

<u>*Step* 1.</u>

We define $G = F$, $H_1 = \emptyset$, $H_2 = \emptyset$ and $f_0(0) =< \#G, 0 >$, where function $F \in \mathcal{R}$ is defined in the following way:

$$\forall x \in \mathbf{N}: \ F(x) = \begin{cases} f_2(x), & \text{if } f_2(x)\downarrow, \\ f_1(x), & \text{if } f_2(x)\uparrow \text{ and } f_1(x)\downarrow, \\ f_0(x), & \text{if } f_2(x)\uparrow \text{ and } f_1(x)\uparrow . \end{cases}$$

It is not hard to see that function F, and thus also G, is recursive. At the same time, in general we can not guarantee that $\#G$ is computable from $\#f_0$, $\#f_1$ and $\#f_2$. However, from construction of functions f_t it will be seen that $\#G$ can be computed if we additionally know, which of the functions f_t is defined for an infinite number of argument values, and which are not.

We denote by $\#_t G$ a number of function G in the case when only functions $f_0, \ldots, f_t$ are defined for an infinite number of argument values.

Step k, $k > 1$.
We define $f_0(k) =< \#_0 G, 0 >$ and $H_1 = H_1 \cup \{(M(F_k[k-1]), k)\}$, $H_2 = H_2 \cup \{(M(F_k[k-1]), k)\}$, where function F_k corresponds to the part of function F that is already defined until the k- th step.

Then for all $t \in \{1,2\}$, for all $(h, k') \in H_t$ we compute $\varphi_h\langle k\rangle(k'+1)$.

If we find $(h,k') \in H_1$, such that $\varphi_h\langle k\rangle(k'+1)\downarrow$, then we define

$$f_1(k'+1) =< \#_1 G\{(k'+1, f_1(k'+1))\}, \varphi_h\langle k\rangle(k'+1)[2]+1 >,$$

define $G = G\{(k'+1, f_1(k'+1))\}$ and $H_2 = \emptyset$.

Similarly, if we find $(h,k') \in H_2$, such that $\varphi_h\langle k\rangle(k'+1)\downarrow$, then we define

$$f_2(k'+1) =< \#_2 G\{(k'+1, f_2(k'+1))\}, \varphi_h\langle k\rangle(k'+1)[2]+1 >,$$

define $G = G\{(k'+1, f_2(k'+1))\}$ and $H_1 = \emptyset$, $H_2 = \emptyset$. After that we go to the next step.
End.

We define the function $f_M = F$. So that function f_0 is total and each of the three functions f_1 and f_2 are defined either for a finite or an infinite number of argument values, then f_M is a total recursive function.

By construction machine M does not $BC[2]$ identify function f_M. At the same time, in each step of our procedure we have equality $F =^2 G$, thus $f_M \in A$.□

However, in general the problem of finding what must be the values of a, n, m in order to have inclusion $BC^a[n] \subset BC[m]$ appears to be quite complicated. The next theorem gives a partial answer to this problem and can be obtained by quite straightforward generalization of proof of the first statement of Theorem 3. At the same time, we do not know whether in this way obtained value $n(a+1)$ gives the smallest possible bound of mind changes, if $a > 0$ or $n > 1$.

Theorem 4 *For all $n \in \mathbf{N}_+$, for all $a \in \mathbf{N}$: $BC^a[n] \subset BC[n(a+1)]$.*□

4 Probabilistic inductive inference of limiting programs

In this section we shall consider probabilistic inference of limiting programs. We shall restrict our attention only to identification without anomalies.

First, the following theorem shows that it is not possible to reduce the number of mind changes that are allowed for each value of identifiable function by lowering the requirements for probability of identification. It can be proved quite similarly as Theorem 1.

Theorem 5 *For all* $p \in (0,1], n \in \mathbf{N}_+ :\ EX[n+1] - BC_p[n] \neq \emptyset$. $\Box$

The next theorem shows that for probabilities $p \in (1/2, 1]$ classes $EX_p[\star]$ form a deterministic hierarchy, which turns out to be the same as we have in the case of FIN identification (see [4]) (i.e. for identification in the case when IIM is allowed to produce only one hypothesis about identifiable function), namely, for each $p \in (1/2, 1]$ class $EX_p[\star]$ is equal to the class $EX_{n/(2n-1)}[\star]$ for some non-zero natural number n. This hierarchy is different from the hierarchy for EX (which is the same as $EX[1]$) identification, where it is known that for each $p \in (0,1]$ class EX_p is equal to class $EX_{1/n}$ for some non zero natural number n (see [6]).

In some sense, the reason, why for $EX[\star]$ we are getting the different hierarchy as for EX, is the fact that here we may be unable to recognize, even in the limit, which hypotheses are wrong. However, it is quite surprising that result turns out to be the same as we have for FIN identification.

Theorem 6 *For all* $n \in \mathbf{N}_+$ *and* $p \in ((n+1)/(2n+1), n/(2n-1)]$ *the following holds:*

1. $EX_{(n+1)/(2n+1)}[\star] - EX_{n/(2n-1)}[\star] \neq \emptyset$, *and*
2. $EX_p[\star] = EX_{n/(2n-1)}[\star]$. $\Box$

Proof. We shall show here the proof only for the case when $n = 2$. For part (2) we are using the method of majority voting, similar to that is used in proof of Theorem 2 from [4], with some adaptations needed to get it work for EX identification. These adaptations can be easily modified also for the general case. For part (1) we are using some kind of diagonalization, which also can be done quite similarly for the general case.

(1) We shall show that there exists set A of total recursive functions, such that $A \in EX_{2/3}[\star]$ and $A \notin EX[\star]$.

We define the set A as the union $A = B \cup C$, where the sets B and C are defined as follows:

$$B = \{f \in \mathcal{R} \mid f = \varphi^\star_{f(0)} \ \&\ f = \varphi^\star_{f(1)}\}$$

and

$$C = \{f \in \mathcal{R} \mid \quad (f = \varphi^\star_{f(0)} \ \vee\ f = \varphi^\star_{f(1)}) \ \&$$
$$(\exists x \in \mathbf{N}, x > 1 :\ [\forall y \in \mathbf{N}, y > x :\ f(x) = f(y) \ \&\ f = \varphi^\star_{f(x)}])\}.$$

Clearly, we have $A \in EX_{2/3}[\star]$, so that A is $EX_{2/3}[\star]$ identifiable by an inductive inference machine M, which for each function f, given for identification, with probabilities $1/3$ outputs each of the sequences $f(0), f(0), \ldots,\ f(1), f(1), \ldots$ and $f(0), f(1), f(2), \ldots$.

It remains to show that $A \notin EX[\star]$.

For each inductive inference machine M we shall construct function $f_M \in A$, such that machine M does not $EX[\star]$ identify function f_M.

We define two recursive functions $f_1, f_2 \in \mathcal{P}_2$ by the following procedure.

Step 1.
For $t \in \{1,2\}$ and for $x \in \{0,1\}$ we define $H_t = 0$ and $f_t(x,0) = \#f_{x+1}$.
(Informally value H_t could be described as the last hypothesis of M produced on function $f_t^\star$).

Step $k, k > 1$.
For $t \in \{1,2\}$ we compute $M(f^\star_{t,k}[k-1])$, where functions $f_{t,k}$ correspond to the parts of functions f_t defined so far.

(i) If for both values $t = 1$ and $t = 2$ we have $H_t = 0$ and $M(f^{\star}_{t,k}[k]) = 0$, then for all $x \geq k+1$ we define $f_1(x,k) = 0$, $f_2(x,k) = 0$ and go to the next step.

(ii) If there exists $t \in \{1,2\}$, such that $H_t = 0$ and $M(f^{\star}_{t,k}[k]) \neq 0$, then for all $x \geq k+1$ we define $f_1(x,k) = \#f_1$, $f_2(x,k) = \#f_2$, define $H_1 = M(f^{\star}_{t,k}[k])$, $H_2 = M(f^{\star}_{t,k}[k])$ and go to the next step.

(iii) If for both values $t = 1$ and $t = 2$ we have $H_t \neq 0$ and $M(f^{\star}_{t,k}[k]) = H_t$, then for all $x \geq k+1$ we define $f_1(x,k) = f_1(x,k-1)$, $f_2(x,k) = f_2(x,k-1)$ and go to the next step.

(iv) If there exists $t \in \{1,2\}$, such that $H_t \neq 0$ and $M(f_{t,k}[k]) \neq H_t$, then for all $s \in \{1,2\}$ we define $H_s = M(f^{\star}_{t,k}[k])$, for all $x \in \mathbf{N}$ define

$$f_s(x,k) = \begin{cases} f_{t,k}(x,k-1), & \text{if } x < k \\ \#f_s, & \text{if } x \geq k \end{cases}$$

and go to the next step.
End.

From the definition of f_1 and f_2 it follows, that for $t \in \{1,2\}$ we have equalities $f_t = \varphi_{f_t(t-1,0)}$. Let for each $t \in \{1,2\}$ function $g_t \in \mathcal{P}$ be defined by equality $g_t(x) = f^{\star}_t$. It is easy to see that $g_1, g_2 \in A$. At the same time, from our construction it follows that at least one of these functions g_{t_0} is not $EX[\star]$ identifiable by machine M, thus we can take $f_M = g_{t_0}$.

(2) Let $p \in (2/3, 1]$. We shall show that for each set $A \in EX_p[\star]$ there exists an inductive inference machine M that $EX[\star]$ identifies every function $f \in A$.

Let $A \in EX_p[\star]$ and let M_p be an inductive inference machine that $EX_p[\star]$ identifies A.
For each $k \in \mathbf{N}$ and for each $f \in A$ we define the set $H(k,f)$ as follows:

$$H(k,f) = \{(h,q) \mid h \in \mathbf{N}, q \in [0,1] : M_p(f[k]) = h \text{ with probability } p\}.$$

For an arbitrary set $S \subset \mathbf{N} \times [0,1]$, such that for each natural number i there exists exactly one number $q \in [0,1]$, such that $(i,q) \in S$, for each $i \in \mathbf{N}$ we define $prob(i,S) = q$, where $q \in [0,1]$ and is such that $(i,q) \in S$.
We define $Prob(S)$ as value $\sum_{i \in \mathbf{N}_+} prob(i,S)$. Thus, meaningfully $Prob(H(k,f))$ is the probability that $M_p(f[k]) \neq 0$.
We shall construct the inductive inference machine M that $EX[\star]$ identifies set A in the following way.

For all functions $f \in A$ machine M initially defines $H = \emptyset$ and $H' = \emptyset$. Then for each value $k \in \mathbf{N}$ machine M computes $H(k,f)$.

(i) If $Prob(H(k,f)) \leq 2/3$, then M will output 0 as hypothesis $M(f[k])$.

(ii) If $H = \emptyset$ and $Prob(H(k,f)) > 2/3$, then it will define $H = H(k,f)$ and $H' = H(k,f)$. Further the machine M will output ξ as hypothesis $M(f[k])$, where function $\xi \in \mathcal{P}_2$, and for all $x, t \in \mathbf{N}$ the value $\xi(x,t)$ is computed by the following procedure P.

For each $(h,q) \in H'$ procedure P will simulate the computation of $\varphi_h(x,t)$, until it will find $y \in \mathbf{N}$, such that

$$\sum_{i \in \mathbf{N}:\ \varphi_i(x,t)=y} prob(i, H') > Prob(H')/2.$$

Then it will output y as value of $\xi(x,t)$.

(iii) If $H \neq \emptyset$ and $Prob(H(k,f) - H) \geq 1/3$, then M will define $H = H(k,f)$ and $H' = H(k,f)$ and will further proceed as in (ii).

(iv) If $H \neq \emptyset$ and $Prob(H(k,f) - H) < 1/3$, and $H' \neq H' \cap H(k,f)$, then M will define $H' = H' \cap H(k,f)$ and will further proceed as in (ii).

(v) If $H \neq \emptyset$ and $Prob(H(k,f) - H) < 1/3$, and $H' = H' \cap H(k,f)$, then M will not change sets H and H' and will further proceed as in (ii).

So that machine M_p $EX_p[\star]$ identifies function f, then only the case (v) can occur more than finite number of times, hence H' will stabilize to a set, which contains numbers of recursive functions, such that more than half of them are correct $EX[\star]$ numbers for f, therefore, also number of function ξ will be a correct $EX[\star]$ number for f, i.e. M will $EX[\star]$ identify each function $f \in A$.$\Box$

A probabilistic hierarchy similar to that for classes $EX_p[\star]$ also holds for classes $EX_p[2]$.

Theorem 7 *For all $n \in \mathbf{N}_+$ and $p \in ((n+1)/(2n+1), n/(2n-1)]$ the following holds:*

1. $EX_{(n+1)/(2n+1)}[2] - EX_{n/(2n-1)}[2] \neq \emptyset$, *and*
2. $EX_p[2] = EX_{n/(2n-1)}[2]$.$\Box$

It seems quite natural to expect that probably the same hierarchy will also take place for all classes $EX[\nu]$, if $\nu \neq 1$. However, this is not the case.

Theorem 8 *For all $n \in \mathbf{N}_+$ and $p \in (n/(n+1), 1]$ the following holds:*

1. $EX_{n/(n+1)}[n] - EX[n] \neq \emptyset$, *and*
2. $EX_p[n] = EX[n]$.$\Box$

Proof. For brevity again we shall show the proof only for part (1) (proof of part (2) is very similar to the proof of part (2) for Theorem 6) and only for the simplest case which is already interesting, i.e. when $n = 3$.

We shall show that there exists set A of total recursive functions, such that $A \in EX_{3/4}[3]$ and $A \notin EX[3]$. We define the set A as the union $A = B \cup C$, where the sets B and C are defined as follows:

$$B = \{f \in \mathcal{R} \mid \forall x \in \{0,1,2\}:\ f = \varphi^3_{f(x)}\}$$

and

$$C = \{f \in \mathcal{R} \mid \quad (\exists x, y \in \{0,1,2\}, x \neq y:\ f = \varphi^3_{f(x)}\ \&\ f = \varphi^3_{f(y)})\ \& \\ (\exists z \in \mathbf{N}:\ [\forall t \in \mathbf{N}, t > z:\ f(z)[1] = f(t)[1]\ \&\ f = \varphi^3_{f(z)[1]}])\},$$

where each value of $f(x)$ we consider as a pair $< f(x)[1], f(x)[2] >$.

Clearly, we have $A \in EX_{3/4}[3]$, so that A is $EX_{3/4}[3]$ identifiable by inductive inference machine M, which for each function f, given for identification, with probabilities $1/4$ outputs each of the sequences $f(0), f(0), \ldots$, $f(1), f(1), \ldots$, $f(2), f(2), \ldots$ and $f(0)[1], f(1)[1], f(2)[1], \ldots$. It remains to show that $A \notin EX[3]$.

We define 4 recursive functions $f_1, \ldots, f_4 \in \mathcal{P}_2$ by the following procedure.

Step 1.
For all $t \in \{1,2,3,4\}$, for all $x \in \{0,1,2,3\}$ we define $f_t(x,0) = \#f_{x+1}$, $c_t = 0$ and define $H = 0$, $c = 0$, $d = 0$.

(Informally H will contain one of the hypotheses produced by M on one of the functions f_t, d will be the the current value of argument for which we shall simulate the computation of φ_H; c and c_t will characterize the number of mind changes for current hypothesis H on argument value d).

Step k, $k > 1$.
For all $t \in \{1,2,3,4\}$ we compute the values $h(t,k) = M(f^3_{t,k}[k+1])$, where functions $f_{t,k}$ correspond to the parts of functions f_t already defined so far.

(i) If $H = 0$ and for all $t \in \{1,2,3,4\}$ we have $h(t,k) = 0$, then for all $s \in \{1,2,3,4\}$ we define $f_s(k+2,0) = 0$ and go to the next step.

(ii) If $H = 0$ and there exists $t \in \{1,2,3,4\}$, such that $h(t,k) \neq 0$, then for all $s \in \{1,2,3,4\}$ we define $f_s(k+2,0) =< \#f_s, s >$, define $H = h(t,k)$, define $d = k+2$ and go to the next step.

(iii) If $H \neq 0$ and for all $t \in \{1,2,3,4\}$ we have $h(t,k) = 0$, then for all $s \in \{1,2,3,4\}$ we define $f_s(k+2,0) = f_s(k+2,0)$ and then compute $\varphi_H\langle k\rangle(d,c)$. If we have $\varphi_H\langle k\rangle(d,c) = y$ for some $y \in \mathbf{N}$, then we find $t \in \{1,2,3,4\}$, such that $f_t(d,c_t) \neq y$, and set $T \subset \{x \in \mathbf{N} \mid 1 \leq x \leq 4, x \neq t\}$, such that $card(T) = 2$, and for all $s \in T$ we have $c_s < 2$. Then for all $s \in T$ we define $f_s(d, c_s+1) = f_t(d,c_t)$ and

- If $c < 2$, then we define $c_s = c_s + 1$, $c = c+1$ and go to the next step;
- If $c = 2$, then we define $c = 0$, $d = k+2$, for all $s \in \{1,2,3,4\}$ define $c_s = 0$ and go to the next step.

(iv) If $H \neq 0$ and there exists $t \in \{1,2,3,4\}$, such that $h(t,k) \neq 0$, then we define $H = h(t,k)$, for all $s \in \{1,2,3,4\}$ define $f_s(d, c_s+1) = f_t(d,c_t)$, define $f_s(k+3,0) =< \#f_s, s >$, define $c_s = 0$, $d = k+2$, $c = 0$ and go to the next step.
End.

From the construction of functions f_t it follows, that for all $t \in \{1,2,3,4\}$ we have equalities $f_t = \varphi_{f_t(t-1,0)}$.

Let for each $t \in \{1,2,3,4\}$ function $g_t \in \mathcal{P}$ be defined by equality $g_t(x) = f^3_t$.

It is easy to see that $g_t \in A$ for all $t \in \{1,2,3,4\}$. At the same time, from construction it follows that at least one of these functions g_{t_0} is not $EX[3]$ identifiable by machine M, so we can take $f_M = g_{t_0}$.□

In addition to these results, we can show that we can improve probability of identification, if we allow to increase the number of mind changes for each value of identifiable function. The proof of this theorem is quite similar to the proof of Theorem 6.

Theorem 9 *For all $m \in \mathbf{N}_+$, with $m \geq 2$, $n \in \mathbf{N}_+$ and $p \in ((n+1)/(2n+1), n/(2n-1)]$ the following holds:*

- $EX_p[m] \subset EX_{n/(2n-1)}[\tilde{n}]$, *where* $\tilde{n} = 2(m-1)$.□

If $m = 2$, then from Theorem 7 it follows that $\tilde{n}$ is the smallest possible value for which this inclusion holds. However, we do not expect that our value of $\tilde{n}$ is the best possible for $m > 2$.

The probabilistic hierarchies for BC identification of limiting programs also turn out to be different from the hierarchy for BC identification of exact programs (which is the same as for EX identification of exact programs) and, besides that, different from hierarchies for EX identification of limiting programs. The BC case also appears to be more easy than EX case. The next theorem gives a complete picture of probabilistic hierarchies for all classes $BC[m]$ and for probabilities from the whole interval $(0,1]$.

Theorem 10 *For all $n \in \mathbf{N}$, $m \in \mathbf{N}_+$ and $p \in (m/(m+n+1), m/(m+n)]$ the following holds:*

1. $BC_{m/(m+n+1)}[m] - BC_{m/(m+n)}[m] \neq \emptyset$, *and*
2. $BC_p[m] = BC_{m/(m+n)}[m]$.□

Proof. Again, we shall show the proof for the one of relatively simple cases, when $m = 2$ and $n = 1$. (The case $m = 1$ gives the already known hierarchy for BC identification of exact programs. Actually, also the case $m = 2$ and $n = 0$ could be already sufficient to show all main ideas of proof, but our choice better illustrates the difference between BC and EX identification.).

In proofs of both parts of this theorem we shall use the following notation.

For each $k \in \mathbf{N}$ and for each $f \in \mathcal{R}$ we define the set $H(k,f)$ as follows:

$$H(k,f) = \{(h,q) \mid h \in \mathbf{N}, q \in [0,1] : \ M_p(f[k]) = h \text{ with probability } p\}.$$

We define values $prob(i,S)$ and $Prob(S)$ similarly as in our proof of part (2) of Theorem 6.

(1) We define set $A \in BC_{2/4}[2] - BC_{2/3}[2]$ in the following way:

$$\begin{array}{ll} A = \{f \in \mathcal{R} \mid & \exists a,b \in \{1,2,3,4\}, a \neq b, \exists s_0, t_0 \in \mathbf{N} : \\ & [\forall s > s_0, \forall x > s : \ f(x) = \varphi^2_{f(s)[a]}(x) \ \& \\ & \forall t > t_0, \forall x > t : \ f(x) = \varphi^2_{f(t)[b]}(x)]\}, \end{array}$$

where we consider each value $f(x)$ as a 5- tuple $< f(x)[1], f(x)[2], f(x)[3], f(x)[4], f(x)[5] >$.

It is easy to see that $A \in BC_{2/4}[2]$, so that each function $f \in A$ is identifiable by IIM that with probability $1/4$ outputs each of the sequences $\#f'_{0,1}, \#f'_{1,1}, \#f'_{2,1}, \ldots, \#f'_{0,2}, \#f'_{1,2}, \#f'_{2,2}, \ldots, \#f'_{0,3}, \#f'_{1,3}, \#f'_{2,3}, \ldots$ and $\#f'_{0,4}, \#f'_{1,4}, \#f'_{2,4}, \ldots$,
where for $i \in \{1,2,3,4\}$, for all $t \in \mathbf{N}$ function $f'_{t,i}$ is defined as

$$\forall x \in \mathbf{N} : \ f_{t,i} = \begin{cases} f(x), & \text{if } x \leq t, \\ \varphi_{f(t)[i]}(x), & \text{if } x > t. \end{cases}$$

We shall show that $A \notin BC_{2/3}[2]$. For each inductive inference machine M we shall construct a function $f_M \in A$, such that machine M does not identify function f_M.

We define 4 recursive functions $f_1, \ldots, f_4 \in \mathcal{P}_2$ by the following procedure.

Step 1.
For all $s \in \{1,2,3\}$ we define $H_s = \emptyset$, and for $t \in \{1,2\}$ define $f_t(0,0) =< \#_1F_1, \#_1F_2, \#_1F_3, \#_1F_4, 1 >$, where functions $F_t \in \mathcal{P}$ are defined in the following way:

$$\forall x \in \mathbf{N}: \; F_t(x) = \begin{cases} f_t(x,1), & \text{if } f_t(x,1) \downarrow, \\ f_t(x,0), & \text{if } f_t(x,1) \uparrow . \end{cases}$$

Similarly, as in proof of Theorem 3, functions F_t can be shown to be recursive, however from $\#f_t$ we can only compute $\#_1F_t$, which is number of F_t in the case when $f_t(x,1) \downarrow$ for only a finite number of values x, and $\#_2F_t$, which is a number of F_t in the case when $f_t(x,1) \downarrow$ for an infinite number of values x.

Step $k,\ k > 1$.
For $t \in \{1,2\}$ we define $f_t(k,0) =< \#_1F_1, \#_1F_2, \#_1F_3, \#_1F_4, 1 >$.

Then for all $s \in \{1,2,3\}$ we define $H_s = H_s \cup \{(H(k, G_k), k)\}$, where function G_k is defined in the following way:

$$\forall x \in \mathbf{N}: \; G_k(x) = \begin{cases} \uparrow, & \text{if } f_{2,k}(x,0) \uparrow, \\ f_{2,k}(x,0), & \text{if } f_{3,k}(x,0) \uparrow, \\ f_{3,k}(x,0), & \text{if } f_{4,k}(x,0) \uparrow, \\ f_{4,k}(x,0), & \text{if } f_{4,k}(x,1) \uparrow, \\ f_{1,k}(x,1), & \text{if } f_{1,k}(x,1) \downarrow, \end{cases}$$

where the functions $f_{t,k}$ as usual correspond to the parts of the functions f_t which are already defined until the k- th step.

Then for all $s \in \{1,2,3\}$ and for all pairs $(H', k') \in H_s$, for all pairs $(h,q) \in H'$ we compute $\varphi_h\langle k\rangle^2(k')$.

If we find $s_0 \in \{1,2,3\}$, $(H_0, k_0) \in H_{s_0}$ and $y \in \mathbf{N}$, such that

$$\sum_{i \in \mathbf{N}:\ \varphi_h\langle k\rangle(k_0)=y} prob(i, H_0) \geq (2/3) \cdot s,$$

then we define

$$H_{s_0} = \{(H', k') \in H_{s_0} \mid k' < k_0\}$$

and

- if $s = 1$, then we define $f_2(k_0,1) =< \#_2F_1, \#_2F_2, \#_2F_3, \#_2F_4, (y + f_2(k_0,0))[5] + 1 >$ and $f_3(k_0,0) =< \#_1F_1, \#_1F_2, \#_1F_3, \#_1F_4, (y + f_2(k_0,0))[5] + 1 >$;
- if $s = 2$, then we define $f_3(k_0,1) =< \#_2F_1, \#_2F_2, \#_2F_3, \#_2F_4, (y + f_3(k_0,0))[5] + 1 >$ and $f_4(k_0,0) =< \#_1F_1, \#_1F_2, \#_1F_3, \#_1F_4, (y + f_3(k_0,0))[5] + 1 >$;
- if $s = 1$, then we define $f_1(k_0,1) =< \#_2F_1, \#_2F_2, \#_2F_3, \#_2F_4, (y + f_4(k_0,0))[5] + 1 >$ and $f_4(k_0,1) =< \#_2F_1, \#_2F_2, \#_2F_3, \#_2F_4, (y + f_4(k_0,0))[5] + 1 >$.

Then we go to the next step.
End.

We define the function f_M as follows:

$$\forall x \in \mathbf{N}: \; F_M(x) = \begin{cases} f_2(x,0), & \text{if } f_3(x,0) \uparrow, \\ f_3(x,0), & \text{if } f_4(x,0) \uparrow, \\ f_4(x,0), & \text{if } f_4(x,1) \uparrow, \\ f_1(x,1), & \text{if } f_1(x,1) \downarrow . \end{cases}$$

We have $f_M \in A$, so that f_M differs only for finitely many argument values from exactly two functions F_t. At the same time, by construction machine M does not identify f_M.

(2) Let $p \in (m/(m+n+1), m/(m+n)] = (2/4, 2/3]$. We shall show that for each set $A \in BC_p[2]$ there exists an inductive inference machine M that $BC_{2/3}[2]$ identifies every function $f \in A$.

Let $A \in BC_p[2]$ and let M_p be an inductive inference machine that $BC_p[2]$ identifies A.
We shall construct the inductive inference machine M that $BC[2]$ identifies set A in the following way.

For each $f \in A$ and for each $k \in \mathbf{N}$ machine M with probability $1/3$ will output each of the sequences $\zeta_1(0), \zeta_1(1), \zeta_1(2), \ldots$, $\zeta_2(0), \zeta_2(1), \zeta_2(2), \ldots$ and $\zeta_3(0), \zeta_3(1), \zeta_3(2), \ldots$, where functions $\zeta_1, \zeta_2, \zeta_3$ are sequentially computed by the following procedure P.

For all $t \in \{1,2,3\}$, for all $k \in \mathbf{N}$ in order to compute $\zeta_t(k)$ procedure P does the following:
At first P computes $H(k,f)$. Then it will output $\#\xi_t$ as the value $\zeta_t(k)$, where functions $\xi_1, \xi_2, \xi_3 \in \mathcal{P}_2$ are defined in the following way:

If $s > 1$ then for all $x \in \mathbf{N}$, for all $t \in \{1,2,3\}$ the value $\xi_t(x,s)$ will be undefined. In order to compute the values $\xi_t(x,0)$ and $\xi(x,1)$ we compute $\varphi_h\langle m\rangle^2(x)$ for all $(h,q) \in H(k,f)$ and for all $m \in \mathbf{N}$.

If $\xi_1(x,0)$ are still undefined and we have found $m_1 \in \mathbf{N}$ and $H_1 \subset H(k,f)$, such that $Prob(H_1) > 2/4$ and for all $(h,q) \in H_1$ we have $\varphi_h\langle m_1\rangle^2(x) = y$ for some $y \in \mathbf{N}$, then we define $\xi_1(x,0) = y$, $\xi_2(x,0) = y$.

If $\xi_1(x,0)$ is already defined and $\xi_3(x,1)$ is still undefined and we have found $m_2 \in \mathbf{N}$ and $H_2 \subset H(k,f)$, such that $Prob(H_2) > 2/4$ and for all $(h,q) \in H_2$ we have $\varphi_h\langle m_2\rangle^2(x) = z$ for some $z \in (\mathbf{N} - \{\xi_1(x,0)\})$, then we define $\xi_3(x,0) = z$, $\xi_1(x,1) = z$.

If $\xi_1(x,1)$ is already defined and $\xi_2(x,1)$ is still undefined and we have found $m_3 \in \mathbf{N}$ and $H_3 \subset H(k,f)$, such that $Prob(H_3) > 2/4$ and for all $(h,q) \in H(k,f)$ we have $\varphi_h\langle m_2\rangle^2(x) = w$ for some $w \in (\mathbf{N} - \{\xi_1(x,0), \xi_1(x,1)\})$, then we define $\xi_2(x,1) = w$, $\xi_3(x,1) = w$.

So that machine M_p $BC_p[2]$ identifies each function $f \in A$, then for all sufficiently large k we have $Prob(\{H \subset H(k,f) \mid \forall (h,q) \in H:\ \varphi_h^2 = f\}) > 2/4$. At the same time, for all $x \in \mathbf{N}$ there can be no more than 3 different values of y, such that

$$Prob(\{H \subset H(k,f) \mid \forall (h,q) \in H:\ \varphi_h^{2,1} = y\ \vee\ \varphi_h^{2,2} = y\}) > 2/4.$$

Therefore, for all sufficiently large k, for all $x \in \mathbf{N}$ we shall have either

- $\xi_1(x,0) = \xi_2(x,0) = f(x)$, or
- $\xi_3(x,0) = \xi_1(x,1) = f(x)$, or
- $\xi_2(x,1) = \xi_2(x,1) = f(x)$,

and thus machine M will $BC[2]$ identify function f.□

If $n = 0$, then Theorem 10 gives the same result for BC identification as we have in Theorem 8 for EX identification. However, as we can see from Theorem 7, in general, probabilistic hierarchies for $EX[2]$ and $BC[2]$ are different. We expect that that will be the case also for other values of $m > 1$.

We have similar result for BC identification as Theorem 9 gives for EX identification. It shows that probability of identification can be improved by increasing the number of allowed mind changes.

Theorem 11 *For all $m, n \in \mathbf{N}_+$ and $p \in (1/(n+1), 1/n]$ the following holds:*

- $BC_p[m] \subset BC_{1/n}[\tilde{n}]$, *where* $\tilde{n} = \lceil (m-1)/n + m \rceil$.□

This result can be proved similarly as the second part of Theorem 10. We do not know whether the bound $\tilde{n} = \lceil (m-1)/n + m \rceil$ is the best possible.

5 Some conclusions and open problems

Here we shall mention some easy noticeable problems that were left unsolved in this paper, but finding solutions of which probably could be interesting.

First, concerning BC identification with anomalies, in general case we still do not know what must be the values of a, b, m, n in order to have inclusion $BC^a[m] \subset BC^b[n]$. Theorems 3 and 4 give only a partial answer to this question.

While we have restricted our attention only to the relations between classes of functions that are identifiable with different probabilities, but with the same number of mind changes, we have obtained complete picture of probabilistic hierarchies for BC identification of limiting programs (Theorem 10). At the same time, this problem is solved only partially in the case of EX identification. However, we expect that for each class $EX[n]$ some generalization of Theorem 7 can be proved by using the same technique, i.e. we expect that in this way it is possible for each value of mind changes n to establish an infinite decreasing probabilistic hierarchy for some interval of probabilities $(p_n, 1]$.

We also do not know anything about $EX[\nu]$ identification with probabilities lower than $1/2$, if $\nu > 1$. However, the similarity of hierarchies for $EX[2]$ and $EX[\star]$ identification with probabilistic hierarchy for FIN identification for probabilities from the interval $(1/2, 1]$ suggests that obtaining such kind of results could be quite hard, similarly as it is for FIN identification, where the situation still is known only partially.

Besides that, it could be interesting to find, whether by increasing the number of mind changes we can increase probability of identification only up to probabilities, given by Theorems 9 and 11 (which corresponds to probabilities from hierarchies for classes $EX[2]$ and $BC[1]$), or we can prove analogous results also for other sequences of probabilities (for example, for EX identification for the sequence $1, 1/2, 1/3, \ldots$ which correspond to the probabilistic hierarchy for classes $EX[1]$).

Finally, one more thing that appears to be interesting is the fact that the same hierarchy, as we have obtained here for $EX[2]$ and $EX[\star]$, holds not only for FIN identification, but also for EX identification up to sets with zero density (see [10]). Therefore, we have the same result in several quite different situations and seemingly for quite different reasons. That in some sense suggests that probably it could be possible to obtain some general result that covers all these (and probably some other) subcases.

6 Acknowledgements

The author wishes to express his gratitude to the Department of Computer and Information Sciences of the University of Delaware and especially to professor John Case for the given possibility to work here and for various kinds of support.

The author of this paper is also very thankful to two anonymous referees for careful reading of the previous version of this paper, for indicating to some errors and for other useful comments.

In particular, due to referees are the notation f^ν, which in several cases makes statements considerably shorter, and the correct statement and proof of the first part of Theorem 3. Also due to them are the idea to simplify the proof of the statement of Corollary 1 by obtaining it as a corollary from simpler result given by Theorem 1.

References

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

[2] J. Case, S. Jain, and A. Sharma. On learning limiting programs. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 193–220, 1992.

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

[4] R. Freivalds. Finite identification of general recursive functions by probabilistic strategies. In *Proceedings of the Conference on Algebraic, Arithmetic and Categorial Methods in Computation Theory FCT'79*, pages 138–145, 1979.

[5] R. Freivalds and J. Vīksna. Inductive inference up to immune sets. In *Proceedings of the International Workshop on Analogical and Inductive Inference AII'89*, volume 397 of *Lecture Notes in Computer Science*, pages 138–147, 1989.

[6] L. Pitt. Probabilistic inductive inference. *Journal of the ACM*, 36:383–433, 1989.

[7] H. Rogers. *Theory of Recursive Functions and Effective Computability.* MIT Press, 1987.

[8] C. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM*, 29:1144–1165, 1982.

[9] R.I. Soare. *Recursively Enumerable Sets and Degrees.* Springer Verlag, 1987.

[10] J. Vīksna. Probabilistic inference of approximations. In *Proceedings of the Conference of Nonmonotonic Logic and Inductive Inference*, volume 659 of *Lecture Notes in Computer Science*, pages 323–332, 1991.

[11] J. Vīksna. Probabilistic inference of limiting functions with bounded number of mind changes. Technical Report LU-IMCS-TRCS-95-1, Institute of Mathematics and Computer Science, University of Latvia, 1995.

[12] R. Wiehagen, R. Freivalds, and E. Kinber. On the power of probabilistic strategies in inductive inference. *Theoretical Computer Science*, 28:111–133, 1984.