Theoretical Computer Science 62 (1988) 289-310
North-Holland

ON THE POWER OF RECURSIVE OPTIMIZERS

Thomas ZEUGMANN
Department of Mathematics, Humboldt University Berlin, 1086 Berlin, German Dem. Rep.

Communicated by R. Karp
Received March 1986
Revised January 1987

Abstract. Problems of the effective synthesis of fastest programs (modulo a recursive factor) for recursive functions given by input-output examples or an arbitrary program are investigated. In contrast to the non-existence result proved by Alton (1974, 1976) we show various existence results. Thereby we deal in detail with the influence of the recursive factor in dependence of the concrete formalization of a fastest program. In particular, we shall show that, even for function classes containing arbitrarily complex functions, the effective synthesis of fastest programs (modulo a simple recursive operator) can be achieved sometimes.

1. Introduction

The present paper deals with the theory of inductive inference, which has been the subject of monographs (cf. [4]) and several survey papers (cf., e.g., [3, 22]), as well as books (cf., e.g., [26]).

In the following we study the question of effectively synthesizing the fastest programs (modulo a recursive factor) for recursive functions. That means, given a class of functions, we ask whether there is a master program, a so-called "recursive optimizer", uniformly synthesizing the fastest programs in an effective way, for any function contained in the considered function class. If we are given any complexity measure in the sense of Blum [6], this problem arises naturally since in any Gödel numbering every recursive function has infinitely many programs and among them there are arbitrarily "bad" ones (i.e., programs possessing an extremely large computational complexity). In [1, 2, 7] the existence problem for such recursive optimizers was firstly studied and generally answered in the negative. On the other hand, the problem of finding programs having nearly minimal size (i.e., an optimal program with respect to a statical complexity measure) has been the subject of intensive research too (cf. [4, 13]). Recently it has been solved completely by Chen [13].

What we present here is a detailed analysis of the existence problem for recursive optimizers. In doing so, we first obtain a sharpened version of Theorem 3.1 pointed out by Alton [1]. Then, in contrast to this negative result, we shall show that, even for function classes containing arbitrarily complex functions, the effective synthesis

0304-3975/88/$3.50

of fastest programs (modulo a simple recursive operator) can be achieved sometimes. Moreover, it will be proved that recursive functions possessing an unbounded range and recursive predicates are extremely different with respect to the existence of fastest programs.

Consequently, the capabilities of recursive optimizers working on recursive predicates are much more restricted as those working on arbitrary recursive functions. In addition, we shall see that the choice of the recursive factor (e.g., recursive functions, computable operators, ...) has great influence on the power of recursive optimizers.

In order to make the paper more readable, we structured it as follows: basic definitions and notations are given in Section 2. Then we present basic results. In Section 4 the main theorems are given. All proofs are contained in Section 5.

2. Basic definitions and notations

Unspecified notation follows Rogers [27]. In addition to or in contrast with [27] we use the following: $\mathbb{N} = \{0, 1, 2, \ldots\}$ denotes the set of all natural numbers. By PF and TF we denote the set of all partial and total functions of one variable over $\mathbb{N}$ respectively.

The class of all partial recursive and recursive functions of one respectively two variables is denoted by P, R, P^2, R^2 respectively. Let $f \in P$, then $\text{Val} f$ denotes the set $\{f(x) \mid x \in \mathbb{N} \text{ and } f(x) \text{ is defined}\}$. $R_{\{0,1\}}$ denotes the set of all $f \in R$ satisfying $\text{Val} f \subseteq \{0, 1\}$ (recursive predicates). Following [23] we define: a measure of computational complexity is a pair (φ, Φ) where φ is an acceptable Gödel numbering of P and $\Phi \in P^2$ satisfies Blum's axioms [6].

Instead of $\lambda x\varphi(i, x)$ we often write φ_i. Let $f \in P$ and $i \in \mathbb{N}$ such that $\varphi_i = f$. Then i is said to be a program for f. For convenience it is sometimes suitable to identify a function from R with the sequence of its values; so 0^i10^∞ denotes the function f with $f(i) = 1$ and $f(n) = 0$ for all $n \neq i$. Let $f \in P$; then $\min_\varphi f$ denotes the least i satisfying $\varphi_i = f$. A function $f \in R$ is called a point of accumulation for a class $U \subseteq R$ iff for all $n \in \mathbb{N}$ there is a function $f' \in U$ with $f'(x) = f(x)$ for all $x \leq n$, but $f' \neq f$. The set of all pairs, triples and finite sequences of natural numbers are denoted by $\mathbb{N}^2$, $\mathbb{N}^3$ and $\mathbb{N}^*$. By $\langle .,. \rangle$, c_3, and c we denote fixed recursive encodings of $\mathbb{N}^2$, $\mathbb{N}^3$ and $\mathbb{N}^*$ onto $\mathbb{N}$ respectively. We write f^n instead of $c(f(0), \ldots, f(n))$ for any $n \in \mathbb{N}$, $f \in P$, where $f(x)$ is defined for all $x \leq n$.

We say that a sequence $(j_n)_{n \in \mathbb{N}}$ of natural numbers converges to a number j iff $j_n = j$ for almost all n. A sequence $(j_n)_{n \in \mathbb{N}}$ is said to be finitely convergent to a number j iff it converges and $j_n = j_{n+1}$, for any n, implies $j_k = j$ for all $k \geq n$.

The abbreviation a.e. ($\forall^\infty$) stands for "almost everywhere" and means "all but finitely many". We write i.o. as an abbreviation for "infinitely often". By NUM we denote the family of function classes $U \subseteq R$ being embeddable in recursively enumerable function classes (i.e., there is a $g \in R$ such that $U \subseteq \{\varphi_{g(i)} \mid i \in N\} \subseteq R$).

For an arbitrary set M we denote the powerset of M by $\mathcal{P}M$. In the sequel $\subset$ denotes a proper set inclusion in contrast to $\subseteq$. Incomparability of sets is denoted by $\#$.

Next we shall consider computable operators. Let $(F_x)_{x\in\mathbb{N}}$ be a canonical enumeration of the finite functions. Following Helm [19] we will distinguish between the following types of computable operators.

A mapping $\mathcal{O}:\mathrm{PF}\to\mathrm{PF}$ is called a *partial recursive operator* iff there exists a recursively enumerable set W such that, for any $y, z\in\mathbb{N}$, it holds that $\mathcal{O}(f)(y)=z$ iff there is an $x\in\mathbb{N}$ such that $c_3(x, y, z)\in W$ and $F_x\subseteq f$. A partial recursive operator is said to be *general recursive* iff $\mathrm{TF}\subseteq$ domain $\mathcal{O}$, and $f\in\mathrm{TF}$ implies $\mathcal{O}(f)\in\mathrm{TF}$.

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

The set of all general recursive operators and total effective operators is denoted by GRO and TEO respectively. See also [27, 32] for more information about these operators.

If $\mathcal{O}$ is an operator which maps functions to functions, we write $\mathcal{O}(f, x)$ to denote the value of the function $\mathcal{O}(f)$ at the argument x. Any computable operator can be realized by a 3-tape Turing machine T which works as follows: If for an arbitrary function $f\in$ domain $\mathcal{O}$, all pairs $(x, f(x))$, $x\in$ domain f, are written down on the input tape of T (repetitions are allowed), then T will write exactly all pairs $(x, \mathcal{O}(f, x))$ on the output tape of T (under unlimited working time).

Let $\mathcal{O}$ be a computable operator from GRO or TEO. Then, for $f\in$ domain $\mathcal{O}$, $m\in\mathbb{N}$, we set: $\Delta\mathcal{O}(f, m)=$ "the least n such that, for all $x\leq n$, $f(x)$ is defined and, for the computation of $\mathcal{O}(f, m)$, the Turing machine T only uses the pairs $(x, f(x))$ with $x\leq n$; if such an n does not exist, we set $\Delta\mathcal{O}(f, m)=\infty$". The functions $h\in R^2$ are sometimes regarded as operators of the form $\mathcal{O}_h(f, x)=h(x, f(x))$. Furthermore, for $r\in R$ we write Ω_r to denote the set of all operators $\mathcal{O}\in\mathrm{GRO}$ with the property that there is a Turing machine T realizing the operator $\mathcal{O}$ such that $\Delta\mathcal{O}(f, m)\leq r(m)$ holds for all functions $f\in$ domain $\mathcal{O}$ and all $m\in\mathbb{N}$.

Now we are able to formalize the concept of a fastest program (modulo a recursive factor). In order to do so, we shall use optimal and weakly optimal programs (cf. [1]) and compression indices (cf. [8, 30]).

Definition 2.1. Let (φ, Φ) be a complexity measure, $\mathcal{O}\in\mathrm{TEO}$, $f\in R$.

(A) Then $i\in\mathbb{N}$ is said to be a *weakly $\mathcal{O}$-optimal program for* f (with respect to (φ, Φ)) iff
 (1) $\varphi_i=f$,
 (2) $\forall^\infty j\,[\varphi_j=f\to\Phi_i(n)\leq\mathcal{O}(\Phi_j, n)$ i.o.].

(B) Also $i\in\mathbb{N}$ is called an *$\mathcal{O}$-optimal program* for f (w.r.t. (φ, Φ)) iff
 (1) $\varphi_i=f$,
 (2) $\forall j\,[\varphi_j=f\to\Phi_i(n)\leq\mathcal{O}(\Phi_j, n)$ a.e.].

Then we also say that the function f is (weakly) $\mathcal{O}$-optimal.

Definition 2.2. Let (φ, Φ) be a complexity measure, $\mathcal{O} \in \mathrm{TEO}$, $f \in R$.

(A) Then $i \in \mathbb{N}$ is said to be an *$\mathcal{O}$-compression index of f* (w.r.t. (φ, Φ)) iff

(1) $\varphi_i = f$,

(2) $\forall j\, [\varphi_j = f \rightarrow \forall n\, \Phi_i(n) \leq \mathcal{O}(\Phi_j, \max\{i, j, n\})]$.

(B) Also $i \in \mathbb{N}$ is called an *absolute $\mathcal{O}$-compression index of f* (w.r.t. (φ, Φ)) iff

(1) $\varphi_i = f$,

(2) $\forall j \forall x\, [\varphi_j(y) = f(y)$ for all $y \leq \Delta\mathcal{O}(\Phi_j, \max\{i, x\}) \rightarrow \Phi_i(x) \leq \mathcal{O}(\Phi_j, \max\{i, x\})]$.

In this case we also say that the function f is (absolutely) $\mathcal{O}$-compressed.

The definitions given above are slightly different with respect to the demands on the "almost-all" quantifier. Moreover, an absolute $\mathcal{O}$-compression index is not only required to satisfy the inequality $\Phi_i(x) \leq \mathcal{O}(\Phi_j, \max\{i, x\})$ for all x and any program j computing the same function as program i, but also for any program j which coincides with program i on a sufficiently large initial segment dependent on i, j and x (i.e., for all arguments $y \leq \Delta\mathcal{O}(\Phi_j, \max\{i, x\})$).

Next we formalize the concept of a recursive optimizer in two different ways. The first approach is directly based on the concept of finite identification introduced by Trakhtenbrot/Barzdin [28] and intensively studied by Lindner [24]. The second one starts from the standardization of programs originally investigated by Kinber [21] and Freivalds/Wiehagen [15].

The main difference between these two concepts is that in the first one only the graph of the function to be identified is successively available, whereas an arbitrary program of the considered function is given as input in the second approach.

Definition 2.3. Let $U \subseteq R$. Then U is said to be *finitely identifiable* ($U \in \mathrm{FIN}$) iff there is a strategy $S \in P$ such that $S(f^n)$ is defined for all $n \in \mathbb{N}$, $f \in U$, and the sequence $(S(f^n))_{n \in \mathbb{N}}$ finitely converges to a program i of f.

By $\mathrm{FIN}(S)$ we denote the family of all those classes U which are finitely identifiable by the strategy S. Note that $\mathrm{FIN} \subset \mathrm{EX}$, where EX denotes the family of all function classes which can be identified in the limit (cf. the definition before the proof of Theorem 4.8).

Now we modify Definition 2.3 by additionally demanding that the identified programs be fastest ones in the sense of Definitions 2.1 and 2.2.

Definition 2.4. Let (φ, Φ) be a complexity measure, $\mathcal{O} \in \mathrm{TEO}$ and $U \subseteq R$. Then U is said to be

(A) finitely *weakly* $\mathcal{O}$-optimally identifiable ($U \in \mathcal{O}$-FINWOPT(φ, Φ)),

(B) *finitely $\mathcal{O}$-optimally identifiable* ($U \in \mathcal{O}$-FINOPT(φ, Φ)),

(C) *finitely $\mathcal{O}$-compressed identifiable* ($U \in \mathcal{O}$-FINCOMP(φ, Φ)),

(D) finitely *absolutely* $\mathcal{O}$-compressed identifiable ($U \in \mathcal{O}$-FINACOMP(φ, Φ))

iff there is a strategy $S \in P$ such that $U \in \mathrm{FIN}(S)$ and for every function $f \in U$ the sequence $(S(f^n))_{n \in \mathbb{N}}$ finitely converges to a number i such that

(A) i is a weakly $\mathcal{O}$-optimal program for f,

(B) i is an $\mathcal{O}$-optimal program for f,

(C) i is an $\mathcal{O}$-compression index of f,

(D) i is an absolute $\mathcal{O}$-compression index of f respectively.

Definition 2.5. Let $U \subseteq R$. U is said to be *finitely standardizable* ($U \in \mathrm{FS}$) iff there is a function $\psi \in P$ such that

(1) $\varphi_i \in U$ implies that $\psi(i)$ is defined and $\varphi_i = \varphi_{\psi(i)}$,

(2) $\varphi_i = \varphi_j$ implies $\psi(i) = \psi(j)$, for all $\varphi_i, \varphi_j \in U$.

By $\mathrm{FS}(\psi)$ we denote the family of all those classes U which are finitely standardizable by the function ψ.

Definition 2.6. Let (φ, Φ) be a complexity measure, $\mathcal{O} \in \mathrm{TEO}$ and $U \subseteq R$. Then U is said to be

(A) finitely *weakly* $\mathcal{O}$-optimally standardizable ($U \in \mathcal{O}$-$\mathrm{FSWOPT}(\varphi, \Phi)$),

(B) *finitely $\mathcal{O}$-optimally standardizable* ($U \in \mathcal{O}$-$\mathrm{FSOPT}(\varphi, \Phi)$),

(C) *finitely $\mathcal{O}$-compressed standardizable* ($U \in \mathcal{O}$-$\mathrm{FSCOMP}(\varphi, \Phi)$),

(D) finitely *absolutely* $\mathcal{O}$-compressed standardizable ($U \in \mathcal{O}$-$\mathrm{FSACOMP}(\varphi, \Phi)$)

iff there is a function $\psi \in P$ such that $U \in \mathrm{FS}(\psi)$ and, for every $\varphi_i \in U$,

(A) $\psi(i)$ is a weakly $\mathcal{O}$-optimal program of φ_i,

(B) $\psi(i)$ is an $\mathcal{O}$-optimal program of φ_i,

(C) $\psi(i)$ is an $\mathcal{O}$-compression index of φ_i,

(D) $\psi(i)$ is an absolute $\mathcal{O}$-compression index of φ_i respectively.

In the sequel we shall investigate the relations between the identification types and standardization types defined above, as well as the dependencies of these types on the complexity measure and the choice of the operator $\mathcal{O}$.

3. Basic results

We start with a negative result.

Theorem 3.1. *Let (φ, Φ) be a complexity measure, $U = \{f \mid f \in R,\ \varphi_{f(0)} = f,\ f(x) = 0 \text{ a.e.}\}$, and let $\mathcal{O} \in \mathrm{TEO}$ be arbitrarily fixed. Then there is no function $\psi \in P$ such that $\varphi_i \in U$ implies that $\psi(i)$ is defined and a weakly $\mathcal{O}$-optimal program for φ_i.*

This theorem is in some sense much stronger than the corresponding non-existence result in Alton [1]. First, there is even a recursive function $h \in R^2$ such that every function from U is absolutely h-compressed (cf. [30]), and not only h-optimal as it was required in [1]. Second, the program equivalence for functions from U is partially decidable, i.e., $U \in \text{FIN}$. So we could prove Alton's conjecture [1] that the undecidability of program equivalence and the non-existence of program optimizers are different things. Third, we only require the optimizer ψ to work on functions from U and not on all h-optimal functions. Finally, the translated program is only demanded to be weakly $\mathcal{O}$-optimal for an operator $\mathcal{O} \in \text{TEO}$ and not to be weakly h'-optimal for some $h' \in R^2$.

As we shall see later, the enlargement of admissible operators from functions $h \in R^2$ to arbitrary operators $\mathcal{O} \in \text{TEO}$ greatly enlarges the family of function classes uniformly having weakly $\mathcal{O}$-optimal programs.

Now one might expect that there is no hope at all to finitely identify or standardize fastest programs, but surprisingly we find the following theorem

Theorem 3.2. *There is a natural complexity measure* (φ, Φ) *such that*

(1) $\mathcal{O}$-FINACOMP(φ, Φ) *contains a function class of infinite cardinality for every operator* $\mathcal{O} \in \text{TEO}$.

(2) $\mathcal{O}$-FINACOMP$(\varphi, \Phi) \subset \mathcal{O}$-FSACOMP$(\varphi, \Phi)$ *for every* $\mathcal{O} \in \text{TEO}$.

On the other hand, Theorem 3.2 does not hold for arbitrary complexity measures. Let $\mathfrak{E}$ denote the identity operator (i.e., $\mathfrak{E}(f) = f$ for all $f \in \text{PF}$). Then we have this theorem.

Theorem 3.3. *There is a complexity measure* (φ, Φ) *such that* $\mathfrak{E}$-FSOPT$(\varphi, \Phi) = \emptyset$.

Considering Theorems 3.2 and 3.3, the problem arises whether results obtained for one complexity measure can be generalized to any measure in a qualitative sense. This question is partially answered by our next theorem.

Theorem 3.4. *Let* (φ, Φ) *and* (φ^*, Φ^*) *be complexity measures and let* $\mathcal{O} \in \text{TEO}$. *Then there is an operator* $\mathcal{O}^* \in \text{GRO}$ *such that*

(1) $\mathcal{O}$-FINWOPT$(\varphi, \Phi) \subseteq \mathcal{O}^*$-FINWOPT$(\varphi^*, \Phi^*)$,

(2) $\mathcal{O}$-FINOPT$(\varphi, \Phi) \subseteq \mathcal{O}^*$-FINOPT$(\varphi^*, \Phi^*)$,

(3) $\mathcal{O}$-FSWOPT$(\varphi, \Phi) \subseteq \mathcal{O}^*$-FSWOPT$(\varphi^*, \Phi^*)$,

(4) $\mathcal{O}$-FSOPT$(\varphi, \Phi) \subseteq \mathcal{O}^*$-FSOPT$(\varphi^*, \Phi^*)$.

Unfortunately, till now we have not been able to extend Theorem 3.4 to $\mathcal{O}$-FINCOMP(φ, Φ), $\mathcal{O}$-FINACOMP(φ, Φ), $\mathcal{O}$-FSCOMP(φ, Φ), and $\mathcal{O}$-FSACOMP(φ, Φ).

On the other hand, it would be interesting to know whether the recursive translation of (weakly) $\mathcal{O}$-optimal progams into (weakly) $\mathcal{O}^*$-optimal programs could be improved. Several interesting results concerning this problem can be found in [17]. In particular, it has been shown there that there are complexity measures (φ, Φ) and (φ^*, Φ^*) having the following properties:

(1) for every operator $\mathcal{O} \in \text{GRO}$ and every function $f \in R$, function f possesses an $\mathcal{O}$-optimal program w.r.t. (φ, Φ) if and only if it has an $\mathcal{O}$-optimal program w.r.t. (φ^*, Φ^*); and

(2) there is no recursive function translating $\mathcal{O}$-optimal programs w.r.t. (φ, Φ) into $\mathcal{O}$-optimal programs w.r.t. (φ^*, Φ^*) for any function $f \in R$.

Corollary 3.5. *Let* (φ, Φ) *and* (φ^*, Φ^*) *be complexity measures, and* $h \in R^2$. *Then there is a function* $h^* \in R^2$ *such that* h-FINWOPT$(\varphi, \Phi) \subseteq h^*$-FINWOPT$(\varphi^*, \Phi^*)$ *and the analogous statement is true for* h-FINOPT(φ, Φ), h-FSWOPT(φ, Φ), *and* h-FSOPT(φ, Φ).

Next, it can be shown that there is no best operator $\mathcal{O}^* \in$ TEO in the sense that $\mathcal{O}^*$-FINOPT$(\varphi, \Phi) \supseteq \mathcal{O}$-FINOPT$(\varphi, \Phi)$ for every operator $\mathcal{O} \in$ TEO (as well as for the other families introduced above) since we have the following result.

Theorem 3.6. *Let* (φ, Φ) *be a complexity measure. Then, for every operator* $\mathcal{O} \in$ TEO, *there is an operator* $\mathcal{O}^* \in$ TEO, *effectively constructable, such that* $\mathcal{O}$-FINOPT$(\varphi, \Phi) \subset \mathcal{O}^*$-FINOPT$(\varphi, \Phi)$.

Finally in this section, we want to investigate the question whether the problem $\mathcal{O}$-FINOPT$(\varphi, \Phi) \neq \emptyset$ is decidable for a fixed measure (φ, Φ) and any operator $\mathcal{O} \in$ TEO.

Theorem 3.7. *Let* (φ, Φ) *be a complexity measure. There is no algorithm deciding for every operator* $\mathcal{O} \in$ TEO *whether or not* $\mathcal{O}$-FINOPT$(\varphi, \Phi) \neq \emptyset$ *iff there exists some operator* $\mathcal{O} \in$ TEO *at all such that* $\mathcal{O}$-FINOPT$(\varphi, \Phi) = \emptyset$.

4. Main results

In this section we start by clarifying the relations between finitely optimal identification and finitely optimal standardization. The next theorem was suggested by Freivalds [14].

Theorem 4.1. *Let* (φ, Φ) *be a complexity measure, and* $\mathcal{O} \in$ TEO. *Then,*

(1) $\mathcal{O}$-FINACOMP$(\varphi, \Phi) =$ FIN $\cap\ \mathcal{O}$-FSACOMP(φ, Φ),

(2) $\mathcal{O}$-FINCOMP$(\varphi, \Phi) =$ FIN $\cap\ \mathcal{O}$-FSCOMP(φ, Φ),

(3) $\mathcal{O}$-FINOPT$(\varphi, \Phi) =$ FIN $\cap\ \mathcal{O}$-FSOPT(φ, Φ),

(4) $\mathcal{O}$-FINWOPT$(\varphi, \Phi) =$ FIN $\cap\ \mathcal{O}$-FSWOPT(φ, Φ).

Theorem 4.1 actually shows that finite identification of fastest programs can be decomposed. Instead of directly looking for a fastest program in the identification process, one may finitely synthesize any program for the function to be identified. After doing so, one can translate this program into a fastest one of the desired type.

Now we point out that the intersection on the right-hand side in Theorem 4.1 is not trivial.

Theorem 4.2. (1) *There is a class $U \in \mathrm{FIN}$ which cannot be finitely weakly $\mathcal{O}$-optimally identified for every operator $\mathcal{O} \in \mathrm{TEO}$.*

(2) *For every complexity measure (φ, Φ) there is a function $h \in R^2$ such that h-*FSACOMP$(\varphi, \Phi) - \mathrm{FIN} \neq \emptyset$.

The proof of Theorem 4.2 shows that finite standardization of fastest programs is generally more powerful than finite identification of fastest programs due to topological reasons. In the sequel we shall study the influence of the concrete choice of the set of admissible operators on the capabilities of the finite identification of fastest programs. In order to do this in a very expressive way, we have to distinguish a certain class of complexity measures. We say that a complexity measure (φ, Φ) satisfies property (+) iff, for all $i, x \in \mathbb{N}$, it holds that if $\Phi_i(x)$ is defined, then $\Phi_i(x) \geq \varphi_i(x)$. In the following "id" denotes the identity function (i.e., $\mathrm{id}(x) = x$), and the functions $f(x) = \mathrm{id}(x) + 1$, $g(x) = \mathrm{id}(x) + x + 1$ are denoted by $\mathrm{id}+1$ and $\mathrm{id}+x+1$ respectively.

Theorem 4.3. *Let (φ, Φ) be a complexity measure satisfying property (+). There is a class $U^{(\varphi,\Phi)}$ such that $U^{(\varphi,\Phi)} \in \mathcal{O}^*$-*FINACOMP$(\varphi, \Phi)$ *for an operator $\mathcal{O}^* \in \Omega_{\mathrm{id}+1}$, but there is no operator $\mathcal{O} \in \Omega_{\mathrm{id}}$ such that $U^{(\varphi,\Phi)} \in \mathcal{O}$-*FINACOMP$(\varphi, \Phi)$.

Remark. The latter theorem can be generalized up to an infinite hierarchy; i.e., for any $x \in \mathbb{N}$ there are a class U and an operator $\mathcal{O}^* \in \Omega_{\mathrm{id}+x+1}$ such that $U \in \mathcal{O}^*$-FINACOMP(φ, Φ), but for every operator $\mathcal{O} \in \Omega_{\mathrm{id}+x}$ it follows $U \notin \mathcal{O}$-FINACOMP(φ, Φ).

The next theorem particularly shows that even operators from $\Omega_{\mathrm{id}+1}$ are much more powerful in generating classes of functions having fastest programs than recursive functions from R^2.

Theorem 4.4. *Let (φ, Φ) be a complexity measure having property (+). There is a class $U^{(\varphi,\Phi)} \in \mathcal{O}^*$-*FINACOMP$(\varphi, \Phi)$ *for some operator $\mathcal{O}^* \in \Omega_{\mathrm{id}+1}$ such that for every function $h \in R^2$ there is a function from $U^{(\varphi,\Phi)}$ that is not weakly h-optimal. In particular, $U^{(\varphi,\Phi)} \notin h$-*FINWOPT$(\varphi, \Phi)$ *for every $h \in R^2$.*

If we restrict ourselves to 0–1 valued functions, the situation changes considerably. As Theorem 4.4 points out, there are classes of functions uniformly having absolute $\mathcal{O}^*$-compression indices for some operator $\mathcal{O}^* \in \Omega_{\mathrm{id}+1}$, and containing arbitrarily complex functions. Thus, $U^{(\varphi,\Phi)}$ from Theorem 4.4 cannot be contained in NUM.

On the other hand, the next theorem shows that classes of recursive predicates being finitely absolutely $\mathcal{O}$-compressed standardizable are in general contained in NUM if we restrict ourselves to operators from Ω_r. Hence such classes of recursive predicates cannot contain arbitrarily complex functions.

Theorem 4.5. *Let (φ, Φ) be a complexity measure, $r \in R$, and let $U \in \mathcal{O}$-*$\textsc{Fsacomp}(\varphi, \Phi) \cap \mathcal{P}R_{\{0,1\}}$ *for some operator $\mathcal{O} \in \Omega_r$. Then there is a function $h \in R^2$ such that $U \in h$-*$\textsc{Fsopt}(\varphi, \Phi)$*, and, moreover, $U \in \mathrm{NUM}$.*

We conjecture that Theorem 4.5 remains valid even if we replace "$\mathcal{O} \in \Omega_r$" by "$\mathcal{O} \in \mathrm{GRO}$". Moreover, it can actually be shown that, for every class $U \subseteq R_{\{0,1\}}$ of functions uniformly having absolute $\mathcal{O}$-compression indices for some operator $\mathcal{O} \in \Omega_r$, there is even a function $h \in R^2$ such that every function from U is absolutely h-compressed (cf. [30, Theorem 4.1]). Furthermore, it remains open whether in Theorem 4.5 the assertion that $U \in h$-$\textsc{Fsopt}(\varphi, \Phi)$ can be sharpened to $U \in h$-$\textsc{Fsacomp}(\varphi, \Phi)$. Nevertheless, Theorems 4.3–4.5 show that recursive predicates and arbitrary functions are extremely different with respect to the existence of absolute compression indices.

With our next theorem we investigate the capabilities of the finite identification, respectively finite standardization of h-compression indices. Please note that the proof of the next theorem uses an idea developed by Jantke [20].

Theorem 4.6. *There are a complexity measure (φ^*, Φ^*) and a function $h^* \in R^2$ such that h^*-*$\textsc{Fincomp}(\varphi^*, \Phi^*) \,\#\, \mathrm{NUM}$.

Corollary 4.7. *Let (φ, Φ) be a complexity measure. Then, for all sufficiently large functions $h \in R^2$ it holds that h-*$\textsc{Fsopt}(\varphi, \Phi) \,\#\, \mathrm{NUM}$.

Finally, we compare the capabilities of weakly $\mathcal{O}$-optimal identification and $\mathcal{O}$-compressed standardization. This is done by the following theorem.

Theorem 4.8. *Let (φ, Φ) be a complexity measure with property $(+)$. Then there is a class $U \subseteq R$ and an operator $\mathcal{O}^* \in \Omega_{\mathrm{id}+1}$ such that $U \in \mathcal{O}^*$-*$\textsc{Finwopt}(\varphi, \Phi)$*, but there is no operator $\mathcal{O} \in \mathrm{GRO}$ at all such that $U \in \mathcal{O}$-*$\textsc{Fscomp}(\varphi, \Phi)$.

It is an open problem whether Theorem 4.8 can be improved. We conjecture that $\mathcal{O}$-optimal identification cannot be achieved for every operator $\mathcal{O} \in \mathrm{GRO}$.

Moreover, we conjecture that for sufficiently large operators $\mathcal{O}$ the following strict inclusions hold:

$$\mathcal{O}\text{-}\textsc{Fsacomp}(\varphi, \Phi) \subset \mathcal{O}\text{-}\textsc{Fscomp}(\varphi, \Phi) \subset \mathcal{O}\text{-}\textsc{Fsopt}(\varphi, \Phi) \subset \mathcal{O}\text{-}\textsc{Fswopt}(\varphi, \Phi).$$

5. Proofs

Before proving Theorem 3.1 we point out that total effective and general recursive operators have the same capabilities in forming classes of functions uniformly having (weakly) optimal programs, although they are extremely different in some properties (cf. [19, 32]). For this purpose, let $R_{(\varphi,\Phi)} = \{\Phi_i \mid \varphi_i \in R\}$ for any complexity measure (φ, Φ).

Proposition 5.1. *Let (φ, Φ) be a complexity measure and let $\mathcal{O} \in$ TEO. Then there is an operator $\mathcal{O}' \in$ GRO such that $\mathcal{O}(f, n) = \mathcal{O}'(f, n)$ a.e. for any function $f \in R_{(\varphi,\Phi)}$.*

The proof of this proposition was given in [31] (cf. Lemma 5.1).

Corollary 5.2. *Let (φ, Φ) be a complexity measure and let $\mathcal{O} \in$ TEO. Let $U \subseteq R$ be a class such that every function from U is (weakly) $\mathcal{O}$-optimal. Then there is an operator $\mathcal{O}' \in$ GRO such that every function from U is (weakly) $\mathcal{O}'$-optimal.*

The proof is an immediate consequence of Proposition 5.1.

Now we are ready to prove Theorem 3.1.

Proof of Theorem 3.1. Let $\mathcal{O} \in$ TEO be arbitrarily fixed. By Proposition 5.1 and [25], there is an operator $\mathcal{O}' \in$ GRO satisfying:

- $\mathcal{O}'$ is monotone (i.e., $f, g \in R$ and $f(n) \leq g(n)$ a.e. implies $\mathcal{O}'(f, n) \leq \mathcal{O}'(g, n)$ a.e.); and
- $\mathcal{O}(f, n) \leq \mathcal{O}'(f, n)$ a.e. for every $f \in R_{(\varphi,\Phi)}$.

Let a function $s \in R$ be chosen such that $\{i0^\infty \mid i \in \mathbb{N}\} = \{\varphi_{s(i)} \mid i \in \mathbb{N}\}$, and for every $i \in \mathbb{N}$ there are infinitely j's satisfying $\varphi_{s(j)} = i0^\infty$.

We set $t(n) = \max\{\Phi_{s(i)}(n) \mid i \leq n\}$ Obviously, $t \in R$ and, for every $i \in \mathbb{N}$, $\Phi_{s(i)}(n) \leq t(n)$ a.e.

Suppose there is an optimizer $\psi \in P$ satisfying: $\varphi_i \in U$ implies that $\psi(i)$ is defined and a weakly $\mathcal{O}'$-optimal program for φ_i. Let z be any fixed program of ψ.

The function $g \in R$ is chosen such that, for all $i, x \in \mathbb{N}$,

$$\varphi_{g(i)}(x) = \begin{cases} i & \text{if } x = 0, \\ 0 & \text{if } x \neq \mu k\,[\Phi_z(i) + 1 \leq k,\ \Phi_{\psi(i)}(k) \leq \mathcal{O}'(t, k)], \\ 1 \dot{-} \varphi_{\psi(i)}(x) & \text{otherwise.} \end{cases}$$

Due to the Recursion Theorem (cf. [27]), there is a number b such that $\varphi_{g(b)} = \varphi_b$. We proceed by showing that the optimizer ψ fails on b.

Case 1: *$\psi(b)$ is not defined.* Then we have $\Phi_z(b) > k$ for all $k \in \mathbb{N}$. Hence, $\psi_{g(b)} = \varphi_b = b0^\infty$. Consequently, $\varphi_b \in U$. By our assumption, it follows that $\psi(b)$ is defined, a contradiction.

Case 2: *$\psi(b)$ is defined.* Then there is a $k_0 \in \mathbb{N}$ such that $\Phi_z(b) + 1 \leq k$ for all $k \geq k_0$.

Subcase 2.1: *There is no $k \geq k_0$ such that $\Phi_{\psi(b)}(k) \leq \mathcal{O}'(t, k)$.* Due to our construction, it follows that $\varphi_b = b0^\infty$. Again we get $\varphi_b \in U$. According to our assumption, $\psi(b)$ is a weakly $\mathcal{O}'$-optimal program for φ_b. Hence, for almost all i with $\varphi_{s(i)} = b0^\infty$ we obtain $\Phi_{\psi(b)}(x) \leq \mathcal{O}'(\Phi_{s(i)}, x)$ i.o. Since the operator $\mathcal{O}'$ is monotone, we consequently conclude $\Phi_{\psi(b)}(x) \leq \mathcal{O}'(t, x)$ i.o. This is a contradiction to $\Phi_{\psi(b)}(k) > \mathcal{O}'(t, k)$ for all $k \geq k_0$.

Subcase 2.2: *There is a $k \geq k_0$ satisfying $\Phi_{\psi(b)}(k) \leq \mathcal{O}'(t, k)$.* Let $x' = \mu k\,[\Phi_z(b) + 1 \leq k, \Phi_{\psi(b)}(k) \leq \mathcal{O}'(t, k)]$. By our construction, $\varphi_{\psi(b)}(x')$ is defined. So we have $\varphi_b(x') = 1 \dot{-} \varphi_{\psi(b)}(x')$. Moreover, it again holds that $\varphi_b(0) = b$ and $\varphi_b(x) = 0$

a.e. Hence, $\varphi_b \in U$. In accordance with our assumption, $\psi(b)$ has to be a weakly $\mathcal{O}'$-optimal program for φ_b. Especially the equality $\varphi_b = \varphi_{\psi(b)}$ has to be satisfied, which is a contradiction. This proves the theorem. $\square$

Proof of Theorem 3.2. The wanted complexity measure is defined as follows: φ is a canonical enumeration of all 3-tape Turing machines with input tape, work tape, and output tape. For all i, x, y we define $\Phi_i(x) = y$ if and only if the read-write head visits exactly y cells on the work tape during the computation of the ith machine when x is given as input, provided that $\Phi_i(x)$ is considered to be undefined if the machine loops on a bounded tape segment (cf. [9]). Now let $\mathfrak{z}$ be the "zero operator", i.e., $\mathfrak{z}(f) = 0^\infty$ for every function $f \in \text{PF}$. Obviously, $\mathfrak{z} \in \text{GRO}$. In order to prove assertion (1) of Theorem 3.2 we set $U = \{0^i 10^\infty \mid i \in \mathbb{N}\}$. It actually suffices to prove that $U \in \mathfrak{z}\text{-}\textsc{Finacomp}(\varphi, \Phi)$. Hartmanis and Hopcroft [18] have shown that there is a function $g \in R$ satisfying $\varphi_{g(i)} = 0^i 10^\infty$, and $\Phi_{g(i)}(x) = 0$ for all $i, x \in \mathbb{N}$. Consequently, $g(i)$ is obviously an absolute $\mathfrak{z}$-compression index of $0^i 10^\infty$ (w.r.t. (φ, Φ)). The wanted strategy can easily be defined now. We omit details.

Before proving assertion (2) we recall the fact that a class $U \in \text{FIN}$ cannot possess a point of accumulation (cf. [24]). In the sequel we shall point out that $\mathfrak{z}$-$\textsc{Fsacomp}(\varphi, \Phi)$ even contains a function class U having a point of accumulation $f \in U$. We set $U = \{0^i 10^\infty \mid i \leqslant \min_\varphi 0^i 10^\infty\} \cup \{0^\infty\}$. Freivalds and Wiehagen [15] prove that U is of infinite cardinality. Hence 0^∞ is a point of accumulation. Let $g \in R$ be chosen such that $\{\varphi_{g(i)} \mid i \in \mathbb{N}\} = \{0^i 10^\infty \mid i \in \mathbb{N}\} \cup \{0^\infty\}$, $i \neq j$ implies $\varphi_{g(i)} \neq \varphi_{g(j)}$, and $\Phi_{g(i)}(x) = 0$ for all $i, x \in \mathbb{N}$. Obviously, $g(i)$ is an absolute $\mathfrak{z}$-compression index of $\varphi_{g(i)}$ w.r.t. (φ, Φ). Before defining the wanted optimizer we note that for all $f \in U$ and $b \geqslant \min_\varphi f$ it follows that there is an $x \leqslant b$ with $f(x) = 1$. Therefore we have $f = 0^\infty$ if and only if $f(x) = 0$ for all $x \leqslant b$. Let z be the number such that $\varphi_{g(z)} = 0^\infty$. Then we set

$$\psi(i) = \begin{cases} g(z) & \text{if } \varphi_i(x) \text{ is defined for all } x \leqslant i \text{ and } \varphi_i(0) = \cdots = \varphi_i(i) = 0, \\ g(\mu k\,[\varphi_{g(k)}(x) = \varphi_i(x) \text{ for all } x \leqslant i]) & \text{if } \varphi_i(x) \text{ is defined for all} \\ \quad x \leqslant i \text{ and there is a } j \leqslant i \text{ with } \varphi_i(j) = 1 \text{ and for all} \\ \quad x \leqslant i,\ x \neq j \text{ it holds that } \varphi_i(x) = 0, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

In order to verify that $U \in \mathfrak{z}\text{-}\textsc{Fsacomp}(\varphi, \Phi)(\psi)$, let $f \in U$ be fixed, and let i and j be any fixed programs of f. We have to show that $\psi(i)$ and $\psi(j)$ are defined and equal, and that $\psi(i)$ is an absolute $\mathfrak{z}$-compression index of f w.r.t. (φ, Φ).

Case 1: $f = 0^\infty$. In particular, we get $\varphi_i(0) = \cdots = \varphi_i(i) = 0$ and $\varphi_j(0) = \cdots = \varphi_j(j) = 0$. Therefore it follows that $\psi(i) = \psi(j) = g(z)$. Due to the definition of the function g, we obtain that $g(z)$ is an absolute $\mathfrak{z}$-compression index of f.

Case 2: $f \neq 0^\infty$. Then there exists an $n \in \mathbb{N}$ such that $f = 0^n 10^\infty$. Moreover, since $f \in U$, it must hold that $n \leqslant \min_\varphi 0^n 10^\infty$. On the other hand, it is obvious that $i, j \geqslant \min_\varphi 0^n 10^\infty$. So $\psi(i)$ and $\psi(j)$ are defined due to the second case in the above construction. In accordance with the choice of the function g, there is exactly one k with $\varphi_{g(k)} = 0^n10^\infty$. Consequently, we get $\psi(i) = \psi(j) = g(k)$, and by construction it follows that $g(k)$ is an absolute $\mathfrak{z}$-compression index of 0^n10^∞. This proves the theorem. □

Proof of Theorem 3.3. The proof of this theorem was given in [31] (cf. Theorem 4 and Corollary 5, p. 629). □

Proof of Theorem 3.4. The proof is an immediate consequence of Proposition 5.1 and the proof of Theorem 10 in [31] (cf. pp. 631–632). □

Proof of Corollary 3.5. This can be proved analogously to Theorem 3.4. □

Proof of Theorem 3.6. Let (φ, Φ) be a complexity measure. Let $\mathcal{O} \in$ TEO be any fixed operator. Due to Proposition 5.1 there is an operator $\mathcal{O}' \in$ GRO with

$$\mathcal{O}(f, x) = \mathcal{O}'(f, x) \quad \text{a.e. for every } f \in R_{(\varphi, \Phi)}. \tag{α}$$

Moreover, by the Speed-up Theorem (cf. [25]), there is a function $f \in R$ which is $\mathcal{O}'$-speedable and which can be effectively found. Hence, the function f cannot possess an $\mathcal{O}'$-optimal program (in fact, f does not have a weakly $\mathcal{O}'$-optimal program). Thus we get $\{f\} \notin \mathcal{O}'$-FINOPT$(\varphi, \Phi)$, and consequently, by (α), $\{f\} \notin \mathcal{O}$-FINOPT(φ, Φ). Now let j be any fixed program of the function f. The wanted operator $\mathcal{O}^*$ is defined in two steps. First we set

$$\mathcal{O}_1(t, x) = \Phi_j(x) + t(x) \quad \text{for every } t \in \text{PF and all } x \in \mathbb{N}.$$

Obviously, $\mathcal{O}_1 \in$ GRO and $\{f\} \in \mathcal{O}_1$-FINOPT(φ, Φ). Then $\mathcal{O}^*(t, x) = \max\{\mathcal{O}'(t, x), \mathcal{O}_1(t, x)\}$ for every $t \in$ PF and all $x \in \mathbb{N}$. In accordance with the latter definition and (α) it follows

$$\mathcal{O}^*(t, x) \geq \mathcal{O}(t, x) \quad \text{for all } t \in \text{PF and almost all } x \in \mathbb{N}.$$

Hence, we have $\mathcal{O}$-FINOPT$(\varphi, \Phi) \subseteq \mathcal{O}^*$-FINOPT$(\varphi, \Phi)$ and since $\{f\} \notin \mathcal{O}$-FINOPT(φ, Φ), the inclusion is proper. □

Proof of Theorem 3.7. This can be proved analogously to Theorem 6 in [31]. □

Proof of Theorem 4.1. We show only assertion (2) here. The other assertions can be proved in an analogous way.

(1) *Necessity*: Let $U \in \mathcal{O}$-FINCOMP$(\varphi, \Phi)(S)$ for some strategy S. Obviously, $U \in$ FIN(S). The wanted optimizer ψ is defined as follows: Let $i \in \mathbb{N}$. If the sequence $(S(\varphi_i^n))_{n \in \mathbb{N}}$ finitely converges to a number j, then we set $\psi(i) = j$. Otherwise, $\psi(i)$ remains undefined. Now let $f \in U$ and let i be any fixed program of f. Hence, $\varphi_i \in R$ and $S(\varphi_i^n)$ is defined for all $n \in \mathbb{N}$. Due to the assumption, the sequence $(S(\varphi_i^n))_{n \in \mathbb{N}}$ converges to a number j and j is an $\mathcal{O}$-compression index of f w.r.t. (φ, Φ). Moreover, since $S(\varphi_i^n) = S(\varphi_b^n)$ for all $n \in \mathbb{N}$ and every program b of f, we obtain: $U \in \mathcal{O}$-FSCOMP$(\varphi, \Phi)(\psi)$.

(2) *Sufficiency*: Let $U \in \mathrm{FIN}(S)$ and $U \in \mathcal{O}$-FSCOMP$(\varphi, \Phi)(\psi)$. The wanted strategy S' satisfying $U \in \mathcal{O}$-FINCOMP$(\varphi, \Phi)(S')$ is defined as follows:

$$S'(f^n) = \begin{cases} S(f^n) & \text{if } n = 0 \text{ or } n > 0 \text{ and } n < \mu x\,[S(f^{x-1}) = S(f^x)], \\ \psi(S(f^n)) & \text{if } n > 0 \text{ and } n \geq \mu x\,[S(f^{x-1}) = S(f^x)]. \end{cases}$$

It remains to verify that the strategy S' satisfies the desired requirements. In order to do so, let $f \in U$. We have to prove that the sequence $(S'(f^n))_{n\in\mathbb{N}}$ finitely converges to an $\mathcal{O}$-compression index of f w.r.t. (φ, Φ). Since $f \in U$, it follows that $S(f^n)$ is defined for all $n \in \mathbb{N}$. Furthermore, the sequence $(S(f^n))_{n\in\mathbb{N}}$ finitely converges. As long as the function f is not identified by the strategy S, the strategy S' works exactly as S. Let now $n = \mu x\,[S(f^{x-1}) = S(f^x)]$. Due to the assumption that $U \in \mathrm{FIN}(S)$, it follows that $\varphi_{S(f^n)} = f$. Consequently, $S'(f^n) = \psi(S(f^n))$ is defined. Moreover, $\psi(S(f^n))$ is an $\mathcal{O}$-compression index of f w.r.t. (φ, Φ). Since $S(f^{k-1}) = S(f^k)$ for all $k \geq n$, we obtain $S'(f^{k-1}) = S'(f^k)$ for all $k > n$. This proves the theorem. □

Proof of Theorem 4.2. Assertion (1) directly follows from the proof of Theorem 3.1 since the class $U = \{f \mid f \in R, \varphi_{f(0)} = f, f(x) = 0 \text{ a.e.}\}$ is obviously contained in FIN.

The proof of assertion (2) is quite similar to that of Theorem 3.2, part (2). The only difficulty we have to overcome is the generalization to any complexity measure. For this purpose let (φ, Φ) be an arbitrarily fixed complexity measure. We again set $U = \{0^i 1 0^\infty \mid i \leq \min_\varphi 0^i 1 0^\infty\} \cup \{0^\infty\}$. Since U is infinite, one obtains that 0^∞ is a point of accumulation. Therefore, $U \notin \mathrm{FIN}$.

It remains to show that there is a function $h \in R^2$ such that $U \in h$-FSACOMP(φ, Φ). Let $g \in R$ be chosen such that $\{\varphi_{g(i)} \mid i \in \mathbb{N}\} = \{0^i 1 0^\infty \mid i \in \mathbb{N}\} \cup \{0^\infty\}$, $i \neq j$, implies $\varphi_{g(i)} \neq \varphi_{g(j)}$, and $g(i) \geq i$ for all $i \in \mathbb{N}$. In order to define the wanted function h, we proceed as follows: Set $t(n) = \max\{\Phi_{g(i)}(x) \mid i \leq n, x \leq n\}$ and define $h(n, m) = t(n) + m$ for all $n, m \in \mathbb{N}$.

Claim. *Let $f \in U$ and $i^* = \mu i\,[\varphi_{g(i)} = f]$. Then $g(i^*)$ is an absolute h-compression index of f w.r.t. (φ, Φ).*

Let $x \in \mathbb{N}$. We set $m = \max\{g(i^*), x\}$. Consequently, $m \geq g(i^*) \geq i^*$ and $m \geq x$. Due to the definition of the function t, we obtain:

$$\begin{aligned} \Phi_{g(i^*)}(x) &\leq \max\{\Phi_{g(i^*)}(z) \mid z \leq m\} \\ &\leq \max\{\Phi_{g(i)}(z) \mid i \leq m,\ z \leq m\} \\ &= t(m) = t(\max\{g(i^*), x\}). \end{aligned}$$

Let j be any number satisfying $f(y) = \varphi_j(y)$ for all $y \leq \max\{g(i^*), x\} = m$. Then we have

$$\Phi_{g(i^*)}(x) \leq t(m) \leq t(m) + \Phi_j(m) = h(m, \Phi_j(m)),$$

i.e., $g(i^*)$ is an absolute h-compression index of f w.r.t. (φ, Φ). Let z be the number satisfying $\varphi_{g(z)} = 0^\infty$.

Let ψ be as in the proof of Theorem 3.2. In the proof of Theorem 3.2, it was shown that $U \in \mathfrak{z}$-FSACOMP$(\varphi, \Phi)(\psi)$, where $\mathfrak{z}$ was the operator that always returned the everywhere-zero function. Clearly, $U \in h$-FSACOMP$(\varphi, \Phi)(\psi)$. □

Proof of Theorem 4.3. The desired function class is defined as follows: $U^{(\varphi,\Phi)} = \{f \mid f \in R,\ \varphi_{f(0)} = f,\ \forall x\, \Phi_{f(0)}(x) \leq f(x+1)\}$. This class was first studied in [11]. Now it suffices to set $\mathcal{O}^*(f, x) = \max\{f(z) \mid z \leq x+1\}$. Obviously, $\mathcal{O}^* \in \Omega_{\mathrm{id}+1}$. Due to property (+), one directly obtains that $f(0)$ is an absolute $\mathcal{O}^*$-compression index of f for every function $f \in U^{(\varphi,\Phi)}$. So $U^{(\varphi,\Phi)}$ can trivially be identified in the sense of $\mathcal{O}^*$-FINACOMP.

The second part of the proof is only sketched here. All the details can be found in [30] (cf. Theorem 4.2, p. 587, and Lemma 3.9, pp. 578–579). The main idea of this proof works as follows. First, one shows that every class of functions uniformly having absolute $\mathcal{O}$-compression indices for some operator $\mathcal{O} \in \Omega_{\mathrm{id}}$ can be identified by a strategy $S \in R$ working consistently on all function $f \in R$ (i.e., $U \in R$-CONS, cf. [30, p. 562]). Then it can be proved that the class $U^{(\varphi,\Phi)}$ defined above cannot belong to R-COMS. So we get the result that for every operator $\mathcal{O} \in \Omega_{\mathrm{id}}$ there is a function from $U^{(\varphi,\Phi)}$ which is not absolutely $\mathcal{O}$-compressed. This yields the desired assertion. □

Proof of Theorem 4.4. We set again $U^{(\varphi,\Phi)} = \{f \mid f \in R,\ \varphi_{f(0)} = f,\ \forall x\, \Phi_{f(0)}(x) \leq f(x+1)\}$. So the first part follows from Theorem 4.3.

Note that the second part is proved using an idea and the proof techniques explained in [29]. This part of the proof is measure-independent. Let (φ, Φ) be fixed. Let $\lambda_i F_i$ be a canonical enumeration of all finite functions exactly defined at an initial segment $\{0, 1, \ldots, n\}$ of natural numbers. Additionally, let the functions $s \in R^2$ and $p \in R^2$ be chosen such that

$$\varphi_{s(i,x)}(y) = \varphi_i(\langle x, y\rangle) \quad \text{for every } i,\ x, \text{ and } y,$$

and

$$\varphi_{p(i,v)}(x) = \begin{cases} F_v(x) & \text{if } x \in \operatorname{Arg} F_n, \\ \varphi_i(x) & \text{otherwise.} \end{cases}$$

Here s is the s_1^1 function of [27] and p is a table patching function.

For the following let $h \in R^2$ be arbitrary and, without loss of generality, let h be monotone in the second argument. Furthermore, let $r \in R$ be a strongly monotone function satisfying $\Phi_i = \varphi_{r(i)}$ for all $i \in \mathbb{N}$. Hence $\operatorname{Val} r$ is recursive. Moreover, for any $j \in \mathbb{N}$ there is at most one $k \in \mathbb{N}$ such that $r(k) = j$. We define $C_{0,0}^j = \emptyset$ for all $j \in \mathbb{N}$. For every $n, x \geq 0$, and for every $j \in \mathbb{N}$ we set $C_{n,x}^j = \emptyset$ if $j \notin \operatorname{Val} r$; otherwise (i.e., there is a $k \in \mathbb{N}$ such that $r(k) = j$) define

$$C_{n,x}^j = \left\{ i \,\middle|\, n \leq i < x,\ i \notin \bigcup_{y<x} C_{n,y}^j,\ \Phi_i(x) \leq \max_{v<x} h(x, \Phi_{p(s(k,i+1),v)}(x)) \right\}.$$

Now let the function $g \in R$ be chosen such that, for every $n, x, j \in \mathbb{N}$,

$$\varphi_{g(j)}(\langle n, 0\rangle) = \begin{cases} s(k, n) & \text{if there is a } k \text{ with } r(k) = j, \\ 0 & \text{otherwise} \end{cases}$$

and, for $x > 0$,

$$\varphi_{g(j)}(\langle n, x\rangle) = \begin{cases} 1 + \max\{\varphi_i(x) \mid i \in C^j_{n,x}\} + \Phi_{s(k,0)}(x-1) \\ \quad \text{if there is a } k \text{ such that } r(k) = j, \\ 0 \quad \text{otherwise.} \end{cases}$$

In accordance with our construction we obtain that $\varphi_{g(j)} \in P$ for every $j \in \mathbb{N}$. Due to the Recursion Theorem [27], there is a number b such that $\varphi_{g(r(b))} = \varphi_b$. In the sequel we shall prove that $\varphi_b \in R$, $\lambda x \varphi_b(\langle 0, x\rangle) \in U^{(\varphi, \Phi)}$ and that $\lambda x \varphi_b(\langle 0, x\rangle)$ is not weakly h-optimal.

Claim 1. $\varphi_b \in R$.

Since $\langle .,. \rangle$ is a recursive encoding of $\mathbb{N}^2$ to $\mathbb{N}$, it suffices to show $\varphi_b(\langle n, x\rangle)$ is defined for all $n, x \in \mathbb{N}$. First we show inductively that $\varphi_b(\langle 0, x\rangle)$ is defined for every $x \in \mathbb{N}$. Due to the construction it holds that

$$\varphi_b(\langle 0, 0\rangle) = \varphi_{g(r(b))}(\langle 0, 0\rangle) = s(b, 0).$$

Since $s \in R^2$, we get that $\varphi_b(\langle 0, 0\rangle)$ is defined. Let now $\varphi_b(\langle 0, x-1\rangle)$ be defined. We have to show that $\varphi_b(\langle 0, x\rangle)$ is defined. By construction we obtain

$$\varphi_b(\langle 0, x\rangle) = 1 + \max\Big\{\varphi_i(x) \,\Big|\, i < x,\ i \notin \bigcup_{y<x} C^{r(b)}_{0,y},\ \Phi_i(x) \le \max_{v<x} h(x, \Phi_{p(s(b,i+1),v)}(x))\Big\} + \Phi_{s(b,0)}(x-1).$$

By assumption, $\varphi_b(\langle 0, x-1\rangle)$ is defined. Therefore, $\Phi_{s(b,0)}(x-1)$ must be defined. It remains to show that $C^{r(b)}_{0,y}$ is computable for all $y < x$, and that $\Phi_{p(s(b,i+1),v)}(x)$ is defined for every $i < x$ satisfying $i \notin \bigcup_{y<x} C^{r(b)}_{0,y}$ and every $v < x$. However, for this purpose it suffices to show that $\Phi_{p(s(b,1),v)}(x), \Phi_{p(s(b,2),v)}(x), \ldots, \Phi_{p(s(b,x),v)}(x)$ are defined for every $i < x$. By the choice of the functions p and s, it is enough to verify that $\varphi_b(\langle x, x\rangle), \varphi_b(\langle x-1, x\rangle), \ldots, \varphi_b(\langle 1, x\rangle)$ are defined. Due to the construction, $\varphi_b(\langle x, x\rangle) = 1 + \Phi_{s(b,0)}(x-1)$ since there is no i with $x \le i < x$. Hence, $\varphi_b(\langle x, x\rangle)$ is defined. Furthermore, we get $\varphi_b(\langle x-1, x\rangle) = 1 + \max\{\varphi_i(x) \mid x-1 \le i < x$ and $\Phi_i(x) \le \max_{v<x} h(x, \Phi_{p(s(b,i+1),v)}(x))\} + \Phi_{s(b,0)}(x-1)$ since the only i satisfying $x-1 \le i < x$ is $x-1$, and, obviously, $x-1 \notin \bigcup_{y<x} C^{r(b)}_{x-1,y}$. Consequently, it suffices to show that $\Phi_{p(s(b,x),v)}(x)$ is defined for every $v < x$. Moreover, $\Phi_{p(s(b,x),v)}(x)$ is defined for every $v < x$ iff $\varphi_{p(s(b,x),v)}(x)$ is defined for every $v < x$. On the other hand, $\varphi_{p(s(b,x),v)}(x)$ is defined if $\varphi_b(\langle x, x\rangle)$ is defined, and this has already been proved. In a completely analogous way, $\varphi_b(\langle x-2, x\rangle), \ldots, \varphi_b(\langle 1, x\rangle)$ is shown to be defined now. Hence, $\lambda x \varphi_b(\langle 0, x\rangle) \in R$, and consequently, $\Phi_{s(b,0)} \in R$.

Claim 2. *For all $x \in \mathbb{N}$ the function $\lambda n\varphi_b(\langle n, x\rangle)$ is total.*

By construction we get $\varphi_b(\langle n, 0\rangle) = s(b, n)$, hence it is defined. Furthermore, for $n \geq x > 0$, in accordance with the construction, it follows that $\varphi_b(\langle n, x\rangle) = 1 + \Phi_{s(b,0)}(x \dot{-} 1)$, and thus defined. Now let $\lambda n\varphi_b(\langle n, x'\rangle) \in R$ for every $x' < x$. We have to prove that $\lambda n\varphi_b(\langle n, x\rangle) \in R$. As it has already been proved, $\varphi_b(\langle n, x\rangle)$ is defined for every $n \geq x$. Now let $n < x$. In order to verify that $\varphi_b(\langle n, x\rangle)$ is defined, it suffices to show $\Phi_{p(s(b,i+1),v)}(x)$ is defined for all $i < x$ and $v < x$. This can be done in a completely analogous way as in Claim 1. We omit details. Hence, $\varphi_b \in R$.

Claim 3. $\lambda x\varphi_b(\langle 0, x\rangle) \in U(\varphi, \Phi)$.

First we note that $\varphi_b(\langle 0, 0\rangle) = \varphi_{s(b,0)}(0) = s(b, 0)$. Due to our construction it immediately follows that

$$\varphi_{s(b,0)}(x+1) = \varphi_b(\langle 0, x+1\rangle) > \Phi_{s(b,0)}(x) \quad \text{for all } x \in \mathbb{N}.$$

This yields Claim 3.

Finally we show that the following claim holds.

Claim 4. *$\lambda x\varphi_b(\langle 0, x\rangle)$ is not weakly h-optimal.*

Let $\varphi_i = \lambda x\varphi_b(\langle 0, x\rangle)$. We have to show that there are infinitely many programs z satisfying $\varphi_i = \varphi_z$ and $\Phi_i(x) > h(x, \Phi_z(x))$ a.e. For this purpose, we first note that $C_{0,x}^{r(b)} - \{0, 1, \ldots, n-1\} = C_{n,x}^{r(b)}$. Moreover, it is obvious that for every n there is a u_n such that if $i < n$ and $i \in \bigcup_{y\in\mathbb{N}} C_{0,y}^{r(b)}$, then $i \in \bigcup_{y<u_n} C_{0,y}^{r(b)}$. Thus, for $i < n$ it holds that $i \notin C_{0,y}^{r(b)}$ for $y > u_n$. So we have shown that $C_{0,x}^{r(b)} = C_{n,x}^{r(b)}$ for every $x > u_n$. Due to the construction one now immediately obtains

$$\varphi_b(\langle 0, x\rangle) = \varphi_b(\langle n, x\rangle) \quad \text{for every } n \text{ and } x > u_n. \tag{1}$$

Furthermore, we claim that $\Phi_i(x) > \max_{v<x} h(x, \Phi_{p(s(b,i+1),v)}(x))$ for every $x > i$. In order to see this, suppose the converse. Hence, there is an x' such that $i \in C_{0,x'}^{r(b)}$. Consequently,

$$\begin{aligned}\varphi_b(\langle 0, x'\rangle) = 1 + \max\{\varphi_i(x') \mid\, & i < x' \text{ and } i \notin \textstyle\bigcup_{y<x'} C_{0,y}^{r(b)} \text{ and} \\ & \Phi_i(x') \leq \max_{v<x'} h(x', \Phi_{p(s(b,i+1),v)}(x'))\} \\ + \Phi_{s(b,0)}(x'-1) & \\ > \varphi_i(x') &\end{aligned}$$

which is a contradiction since $\varphi_i = \lambda x\varphi_b(\langle 0, x\rangle)$. Moreover, (1) yields that there is a v_i such that $\varphi_{p(s(b,i+1),v_i)}(x) = \varphi_b(\langle 0, x\rangle)$ for every $x \in \mathbb{N}$. Thus, it suffices to set $z = p(s(b, i+1), v_i)$ to obtain one desired program z. This process can be iterated since h is monotone in the second argument, and so one finds infinitely many programs z with the wanted properties. $\square$

Proof of Theorem 4.5. Let (φ, Φ), $r \in R$ and $\mathcal{O} \in \Omega_r$, be arbitrarily fixed and suppose that $U \in \mathcal{O}$-FSACOMP$(\varphi, \Phi)(\psi) \cap \mathcal{P}R_{\{0,1\}}$. In [30, Theorem 4.1], it has been shown that $U \in$ NUM then. This result can be obtained by proving that, under the above assumptions, U is identifiable by a strategy working reliably on the set of all total functions (i.e., $U \in$ TF-REL). Then it is not hard to show that TF-REL $\cap\, \mathcal{P}R_{\{0,1\}} =$ NUM $\cap\, \mathcal{P}R_{\{0,1\}}$. (For details the reader is referred to [30, pp. 582–583].) Since $U \in$ NUM, there is a function $g \in R$, $g(i) \geq i$ for all i such that $U \subseteq \{\varphi_{g(i)} \mid i \in \mathbb{N}\} \subseteq R$. We define a function t as follows: for all $n \in \mathbb{N}$: $t(n) = \max\{\Phi_{g(i)}(x) \mid i \leq n, x \leq n\}$. As in the proof of Theorem 4.2 it now follows:

$$\Phi_{g(i)}(x) \leq t(\max\{g(i), x\}) \quad \text{for all } x \in \mathbb{N}. \tag{α}$$

Using the operator $\mathcal{O} \in \Omega_r$ we define an operator $\mathcal{O}^*$ as follows: for all $f \in$ TF we set

$$\mathcal{O}^*(f, x) = \max\{\mathcal{O}(p, x) \mid p \in \mathrm{TF}, \forall z \leq r(x)\; p(z) \leq \max\{f(z), x\}\}.$$

Obviously we have $\mathcal{O}^* \in \Omega_r$ and $\mathcal{O}(f, x) \leq \mathcal{O}^*(f, x)$ for every $f \in$ TF and $x \in \mathbb{N}$. Now we set $t^* = \mathcal{O}^*(t)$ and define $h(n, m) = t^*(n) + m$.

Claim. $U \in h$-FSOPT$(\varphi, \Phi)(\psi)$.

For this purpose, recall that for every $\varphi_i \in U$ it holds that $\psi(i)$ is defined and is an absolute $\mathcal{O}$-compression index of φ_i. In particular, for every j satisfying $\varphi_i = \varphi_{g(j)}$ we obtain:

$$\Phi_{\psi(i)}(x) \leq \mathcal{O}(\Phi_{g(j)}, \max\{\psi(i), x\}) \quad \text{for all } x \in \mathbb{N}. \tag{β}$$

Using (α) and the properties of the operator $\mathcal{O}^*$ we get from (β):

$$\Phi_{\psi(i)}(x) \leq \mathcal{O}^*(\Phi_{g(j)}, \max\{\psi(i), x\}) \leq \mathcal{O}^*(t, x) = t^*(x) \quad \text{a.e.} \tag{γ}$$

Let k be any program of φ_i. By (γ) it follows

$$\Phi_{\psi(i)}(x) \leq t^*(x) \leq t^*(x) + \Phi_k(x) = h(x, \Phi_k(x)) \quad \text{a.e.}$$

Thus $\psi(i)$ is an h-optimal program for φ_i. This proves the theorem. □

Before proving Theorem 4.6 we quote the Operator Recursion Theorem discovered by Case [10].

Operator Recursion Theorem. *Let $\mathcal{O}$ be an effective operator. Then there is a strictly increasing function $t \in R$ such that $\mathcal{O}(t, \langle i, x\rangle) = \varphi_{t(i)}(x)$ for all $i, x \in \mathbb{N}$.*

Proof of Theorem 4.6. (1) NUM $- h^*$-FINCOMP$(\varphi^*, \Phi^*) \neq \emptyset$ directly follows from the proof of Theorem 3.1, independently of the complexity measure (φ^*, Φ^*) and $h^* \in R^2$.

(2) There are a complexity measure (φ^*, Φ^*) and a function $h^* \in R^2$ such that h^*-FINCOMP$(\varphi^*, \Phi^*) -$ NUM $\neq \emptyset$.

Let (φ, Φ) be any fixed complexity measure. We set $\varphi^* = \varphi$. Furthermore, we define an operator $\mathcal{O}$ as follows:

$$\mathcal{O}(t, \langle i, x\rangle) = \begin{cases} t(i) & \text{if } x = 0, \\ \Phi_i(x) + \varphi_i(x) + t(i) + 1 & \text{if } x > 0 \end{cases}$$

for all functions $t \in P$. Obviously, $\mathcal{O}$ is effective. Due to the Operator Recursion Theorem there is a strictly increasing function $t^* \in R$ such that

$$\mathcal{O}(t^*, \langle i, x\rangle) = \varphi_{t^*(i)}(x) \quad \text{for every } i, x \in \mathbb{N}.$$

Since the function t^* is strictly increasing, we have that Val t^* is recursive. Now we are ready to define Φ^*: for every j and x we set

$$\Phi_j^*(x) = \begin{cases} 0 & \text{if } j \in \text{Val } t^*,\ x = 0, \\ \Phi_i(x) & \text{if } j \in \text{Val } t^*,\ j = t^*(i),\ x > 0, \\ \Phi_j(x) & \text{otherwise.} \end{cases}$$

Claim 1. (φ^*, Φ^*) *is a complexity measure.*

Case 1: $j \notin \text{Val } t^*$. Due to the construction it holds that $\Phi_j^* = \Phi_j$. So we have $\text{Arg } \varphi_j^* = \text{Arg } \varphi_j = \text{Arg } \Phi_j = \text{Arg } \Phi_j^*$, and, obviously, $\Phi_j(x) = y$ is recursive.

Case 2: $j \in \text{Val } t^*$. Then there is exactly one number i such that $t^*(i) = j$. Hence,

$$\Phi_j^*(x) = \begin{cases} 0 & \text{if } x = 0, \\ \Phi_i(x) & \text{if } x > 0. \end{cases}$$

On the other hand we have

$$\begin{aligned} \varphi_j^*(x) &= \varphi_{t^*(i)}(x) \\ &= \mathcal{O}(t^*, \langle i, x\rangle) = \begin{cases} t^*(i) & \text{if } x = 0, \\ \Phi_i(x) + \varphi_i(x) + t^*(i) + 1 & \text{if } x > 0. \end{cases} \end{aligned}$$

Therefore, $\text{Arg } \varphi_j^* = \text{Arg } \Phi_j^*$ since $\Phi_j^*(x) = \Phi_i(x)$ and $\Phi_i(x)$ is defined iff $\varphi_i(x)$ is defined. Moreover, since Val t^* is recursive, one directly obtains that the predicate "$\Phi_j^*(x) = y$" is recursive for all j, x and y. This proves Claim 1.

Now we define the wanted class U_{Φ^*} as follows:

$$U_{\Phi^*} = \{f \mid f \in R,\ f \text{ is strictly increasing},\ \varphi_{f(0)} = f,\ \forall x\, \Phi^*_{f(0)}(x) \leq f(x)\}.$$

Claim 2. $U_{\Phi^*} \notin \text{NUM}$.

Suppose the converse, i.e., there is a function g such that $U_{\Phi^*} \subseteq \{\varphi_{g(i)} \mid i \in \mathbb{N}\} \subseteq R$. Let $f(0) = \varphi_{g(0)}(0)$, and let $f(x) = \varphi_{g(x)}(x) + f(x-1)$ for all $x > 0$. Obviously, f is strictly increasing. Due to [30] (cf. Lemma 3 in the proof of Theorem 4.1, p. 584) there is a program i such that $\varphi_i = f$ and Φ_i is also strictly increasing. Since

$$\varphi_{t^*(i)}(x) = \begin{cases} t^*(i) & \text{if } x = 0, \\ \Phi_i(x) + f(x) + t^*(i) + 1 & \text{if } x > 0, \end{cases}$$

we obtain that $\varphi_{t^*(i)}(0) = t^*(i)$ and that $\varphi_{t^*(i)}$ is strictly increasing. Furthermore, since

$$\Phi^*_{t^*(i)}(x) = \Phi_i(x) \leq \Phi_i(x) + \varphi_i(x) + t^*(i) + 1 = \varphi_{t^*(i)}(x), \quad x > 0$$

and

$$\Phi^*_{t^*(i)}(0) = 0 \leq t^*(i) = \varphi_{t^*(i)}(0),$$

it follows that $\varphi_{t^*(i)} \in U_{\Phi^*}$.

In accordance with our assumption, there is at least one k such that $\varphi_{t^*(i)} = \varphi_{g(k)}$. On the other hand, we get

$$\varphi_{t^*(i)}(k) = \Phi_i(k) + f(k) + t^*(i) + 1 \neq \varphi_{g(k)}(k),$$

a contradiction. This proves Claim 2.

Claim 3. *There is a function $h^* \in R^2$ such that $f(0)$ is an h^*-compression index for every function $f \in U_{\Phi^*}$.*

We define: $h^*(n, m) = \max_{i \leq n}\{\varphi_i(n) \mid \Phi^*_i(n) \leq m\}$. According to this definition we have $\varphi_i(n) \leq h^*(n, \Phi^*_i(n))$ for all i and $n \geq i$. Let now $f \in U_{\Phi^*}$ and let j be any program of f. We set $m = \max\{f(0), j, x\}$. Consequently, $m \geq j$, $m \geq x$. Since the function f is strictly increasing and satisfies $\Phi^*_{f(0)}(x) \leq \varphi_{f(0)}(x)$ for all $x \in \mathbb{N}$ we obtain

$$\Phi^*_{f(0)}(x) \leq \varphi_{f(0)}(x) = f(x) = \varphi_j(x) \leq \varphi_j(m) \leq h^*(m, \Phi^*_j(m)),$$

and thus $f(0)$ is an h^*-compression index of f w.r.t. (φ^*, Φ^*). Now let f be given as input. Then the wanted strategy S outputs $f(0)$. Hence, $U_{\Phi^*} \in h^*\text{-FINCOMP}(\varphi^*, \Phi^*)$. □

Proof of Corollary 4.7. The proof is an immediate consequence of Theorem 4.6, the fact that

$$h\text{-FINCOMPT}(\varphi, \Phi) \subseteq h\text{-FINOPT}(\varphi, \Phi) \subseteq h\text{-FSOPT}(\varphi, \Phi),$$

and Theorem 3.3 as well as Theorem 3.1. □

Before proving Theorem 4.8 we define identification in the limit.

Definition. Let $U \subseteq R$. Then U is called *identifiable in the limit* if there is a strategy $S \in P$ such that for every function $f \in U$ and every $n \in \mathbb{N}$ the value $S(f^n)$ is defined and the sequence $(S(f^n))_{n \in \mathbb{N}}$ converges in the limit to a number j such that $\varphi_j = f$. Let EX denote the family of all classes $U \subseteq R$ identifiable in the limit. The identification type EX is sometimes also denoted by LIM (e.g., [30, 31]) or GN (e.g., [4]).

Proof of Theorem 4.8. We define: $U = \{f \mid f \in R,\ \varphi_{f(0)} = f,\ \Phi_{f(0)}(x) \leq f(x+1) \text{ i.o.}\}$. Then $\mathcal{O}^*$ is exactly the same operator as in the proof of Theorem 4.3. Then $U \in \mathcal{O}^*\text{-FINWOPT}(\varphi, \Phi)(S)$, where S is the strategy which, on f, simply outputs $f(0)$. In order to verify that there is no operator $\mathcal{O} \in \text{GRO}$ such that $U \in \mathcal{O}\text{-FSCOMP}(\varphi, \Phi)$, we prove that for every operator $\mathcal{O} \in$ GRO there must be a function from U not being $\mathcal{O}$-compressed. For this purpose, let us suppose the converse, i.e., there is an operator $\mathcal{O}$ such that every function from U has an $\mathcal{O}$-compression index. In [8] it has been proved that then U is identifiable in the limit by a strategy working reliably on R. On the other hand, let $U_0 = \{f \mid f \in R, f(x) = 0 \text{ a.e.}\}$. Obviously, $U_0 \in$ NUM, and hence U_0 is also contained in EX(S) for a strategy even working reliably on R. Moreover, due to the Union Theorem in [8] it follows that $U \cup U_0 \in$ EX. The proof is finished when we show that there is no strategy S identifying $U \cup U_0$ in the limit since this yields a contradiction. This is done by using an idea of Gold [16] which has been refined by Barzdin [4, Chapter I, pp. 82-88]. Nevertheless, the proof might even be shorter by using the techniques of [12].

Suppose, $U \cup U_0 \in$ EX(S), whereas, without loss of generality, $S \in R$. Let the function $r \in R$ be chosen such that $\Phi_i = \varphi_{r(i)}$ for all i and r is strictly monotone. Furthermore, let the function $g \in R$ be chosen such that

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

and proceed as follows: Compute $S(\varphi^0_{g(j)}) = n_1$. Then define $\varphi_{g(j)}(2)$, $\varphi_{g(j)}(3), \ldots,$ to be zero until a k_1 is found such that either

(α) $S(c(\varphi_{g(j)}(0)00^{k_1})) \neq n_1$, or

(β) $S(c(\varphi_{g(j)}(0)10^{k_1})) \neq n_1$.

Since the strategy S is supposed to identify $U \cup U_0$, either (α) or (β) must happen. If (α) happens, then define $\varphi_{g(j)}(1) = 0$, and if (β) happens, then define $\varphi_{g(j)}(1) = 1$. Moreover, we set $\varphi_{g(j)}(k_1+2) = \varphi_j(k_1+1)$ if $\varphi_j(k_1+1)$ is defined. Otherwise, $\varphi_{g(j)}(k_1+2)$ remains undefined and $\varphi_{g(j)}(x)$ will be undefined for all $x \geq k_1+2$. If $\varphi_{g(j)}(k_1+2)$ is defined, we proceed as follows: Compute $S(\varphi^{k_1+2}_{g(j)}) = n_2$. Define $\varphi_{g(j)}(k_1+3), \ldots$ to be zero until a k_2 is found such that either

(α) $S(c(\varphi^{k_1+2}_{g(j)}00^{k_2})) \neq n_2$, or

(β) $S(c(\varphi^{k_1+2}_{g(j)}10^{k_2})) \neq n_2$.

Again, (α) or (β) must happen since the strategy S is supposed to identify in particular U_0. If (α) happens, then set $\varphi_{g(j)}(k_1+3) = 0$, otherwise define $\varphi_{g(j)}(k_1+3) = 1$. Furthermore, we set $\varphi_{g(j)}(k_1+3+k_2) = \varphi_j(k_1+2+k_2)$ if $\varphi_j(k_1+2+k_2)$ is defined. Otherwise, $\varphi_{g(j)}(x)$ remains undefined for all $x \geq k_1+3+k_2$.

By iteration of this construction we define $\varphi_{g(j)}$ completely. Due to the Recursion Theorem there is a number b such that $\varphi_{g(r(b))} = \varphi_b$. By the construction we get $\varphi_b(0) = b$. Let $S(\varphi^0_b) = n_1$ and k_1 may satisfy either

(α) $S(c(b00^{k_1})) \neq n_1$, or

(β) $S(c(b10^{k_1})) \neq n_1$.

Thus $\varphi_b(x)$ is defined for all $x \leq k_1+1$. Furthermore, $\Phi_b(k_1+1) = \varphi_{r(b)}(k_1+1)$, hence defined. So we get

$$\varphi_b(k_1+2) = \varphi_{r(b)}(k_1+1) = \Phi_b(k_1+1).$$

Inductively one now easily shows that $\varphi_b \in U$. On the other hand, the strategy S

changes its hypothesis infinitely often if φ_b is given as input. This proves the theorem. □

References

[1] D.A. Alton, Non-existence of program optimizers in several abstract settings, *J. Comput. System Sci.* **12** (1976) 368-393.

[2] D.A. Alton and J.L. Lowther, Non-existence of program optimizers in an abstract setting, in: *Proc. Symp. on Programming*, Paris, 1974, Lecture Notes in Computer Science **19** (Springer, Berlin, 1974) 96-105.

[3] D. Angluin and C. Smith, A survey of inductive inference: theory and methods, *Comput. Surveys* **15** (1983) 237-269.

[4] Ya.M. Barzdin, ed., *Theory of Algorithms and Programs*; *I, II, III* (Latvian State University, Riga, 1974, 1975, 1976) (in Russian).

[5] Ya.M. Barzdin, Inductive inference of automata, functions and programs, in: *Proc. Internat. Mathematical Congress* (1974) 455-460 (in Russian); or *Amer. Math. Soc. Transl.* **109** (1977) 107-122 (in English).

[6] M. Blum, Machine independent theory of complexity of recursive functions, *J. ACM* **14** (1967) 322-336.

[7] M. Blum, On the size of machines, *Inform. and Control* **11** (1967) 254-265.

[8] L. Blum and M. Blum, Toward a mathematical theory of inductive inference, *Inform. and Control* **28** (1975) 122-155.

[9] A. Borodin, Computational complexity and existence of complexity gaps, *J. ACM* **19** (1972) 158-173.

[10] J. Case, Periodicity in generation of automata, *Math. Systems Theory* **8** (1974) 15-32.

[11] J. Case and S. Ngo-Manguelle, Refinements of inductive inference by popperian machines, Tech. Rep. 152, Dept. of Computer Science, State University of New York at Buffalo, 1979.

[12] J. Case and C. Smith, Comparison of identification criteria for machine inductive inference, *Theoret. Comput. Sci.* **25** (1983) 193-220.

[13] J. Chen, Tradeoffs in the inductive inference of nearly minimal size programs, *Inform and Control* **52** (1982) 68-86.

[14] R.V. Freivalds, Personal communication, 1983.

[15] R.V. Freivalds and R. Wiehagen, Inductive inference with additional information, *J. Inform. Process. and Cybernet.* (*EIK*) **15** (1979) 179-185.

[16] E.M. Gold, Language identification in the limit, *Inform. and Control* **10** (1967) 447-474.

[17] J. Hartmanis, Computational complexity of formal translations, *Math. Systems Theory* **8** (1975) 156-166.

[18] J. Hartmanis and J.E. Hopcroft, An overview of the theory of computational complexity, *J. ACM* **18** (1971) 444-475.

[19] J. Helm, On effectively computable operators, *Z. Math. Logik Grundlag. Math.* **17** (1971) 231-244.

[20] K.P. Jantke, Personal communication, 1985.

[21] E. Kinber, On comparison of limit identification and standardization of general recursive functions (in Russian), in: Ya.M. Barzdin, ed., *Theory of Algorithms and Programs II* (Latvian State University, Riga, 1975) 45-57.

[22] R. Klette and R. Wiehagen, Research in the theory of inductive inference by GDR mathematicians—a survey, *Inform. Sci.* **22** (1980) 149-169.

[23] L.H. Landweber and E.L. Robertson, Recursive properties of abstract complexity classes, *J. ACM* **19** (1972) 296-308.

[24] R. Lindner, Algorithmische Erkennung, Dissertation B, Friedrich-Schiller-Universität, Jena, 1972.

[25] A.R. Meyer and P.C. Fischer, On computational speed-up, in: *Conf. Rec. Ninth Ann. IEEE Symp. on Switching and Automata Theory* (1968) 351-355.

[26] D. Osherson, M. Stob and S. Weinstein, *Systems that Learn* (MIT Press, Cambridge, MA, 1986).

[27] H. Rogers Jr., *Theory of Recursive Functions and Effective Computability* (McGraw-Hill, New York 1967).

[28] B.A. Trakhtenbrot and Ya.M. Barzdin, *Finite Automata-Behavior and Synthesis*, Fundamental Studies in Computer Science 1 (North-Holland, Amsterdam, 1975).

[29] P. Young, Easy constructions in complexity theory, *Proc. Amer. Math. Soc.* 37 (1973) 555–563.

[30] T. Zeugmann, A posteriori characterizations in inductive inference of recursive functions, *J. Inform. Process and Cybernet.* (*EIK*) 19 (1983) 559–594.

[31] T. Zeugmann, On the synthesis of fastest programs in inductive inference, *J. Inform. Process. and Cybernet* (*EIK*) 19 (1983) 625–642.

[32] T. Zeugmann, On the nonboundability of total effective operators, *Z. Math. Logik Grundlag. Math.* 30 (1984) 169–172.