RAIRO-Theor. Inf. Appl. 44 (2010) 379–418
DOI: 10.1051/ita/2010020

Available online at:
www.rairo-ita.org

FUNCTION OPERATORS SPANNING THE ARITHMETICAL AND THE POLYNOMIAL HIERARCHY

Armin Hemmerling[1]

Abstract. A modified version of the classical $\boldsymbol{\mu}$-operator as well as the first value operator and the operator of inverting unary functions, applied in combination with the composition of functions and starting from the primitive recursive functions, generate all arithmetically representable functions. Moreover, the nesting levels of these operators are closely related to the stratification of the arithmetical hierarchy. The same is shown for some further function operators known from computability and complexity theory. The close relationships between nesting levels of operators and the stratification of the hierarchy also hold for suitable restrictions of the operators with respect to the polynomial hierarchy if one starts with the polynomial-time computable functions. It follows that questions around P *vs.* NP and NP *vs.* coNP can equivalently be expressed by closure properties of function classes under these operators. The polytime version of the first value operator can be used to establish hierarchies between certain consecutive levels within the polynomial hierarchy of functions, which are related to generalizations of the Boolean hierarchies over the classes Σ_k^p.

Mathematics Subject Classification. 68Q15, 03D15, 03D55.

1. Introduction and overview

The investigations which led to this paper started with the observation that the first value operator can replace the classical $\boldsymbol{\mu}$-operator in generating the computable functions from the (partial) primitive recursive ones, *cf.* Section 1 in [13].

Keywords and phrases. Arithmetical hierarchy, polynomial hierarchy, Boolean hierarchy, P versus NP, NP versus coNP, first value operator, minimalization, inversion of functions.

[1] Ernst-Moritz-Arndt–Universität Greifswald, Institut für Mathematik und Informatik. Walther-Rathenau-Str. 47, 17487 Greifswald, Germany; hemmerli@uni-greifswald.de

Article published by EDP Sciences

The first value operator in its original form was introduced by Epstein, Haas and Kramer [8] in order to study Ershov's hierarchy [9].

In the present context it turned out that even a special version of this operator is still more powerful than the classical $\boldsymbol{\mu}$ if it is repeatedly applied in connection with the composition of functions: it leads to all arithmetically representable functions. Also a modified version of the $\boldsymbol{\mu}$-operator as well as the operator of inverting unary functions and some further operators known from computability and complexity theory have this power. A more detailed treatment considers the nesting levels (degrees) of these operators and shows that they span the classes of the arithmetical hierarchy in several ways. This is developed in Sections 3-5, after the basic notions have been introduced in Section 2.

Even if these results may contribute some new items from the computability point of view, they surely would not be of big interest if they did not have analogues in complexity theory, in particular concerning polynomial-time complexity. As the arithmetical hierarchy has its polytime counterpart by the polynomial hierarchy, the function operators considered so far can naturally be restricted in such a way that they span the polynomial hierarchy if they are repeatedly applied, starting with polynomial-time computable functions. To be more precise, since we deal with functions and function operators, we always consider hierarchies of (classes of) arithmetical functions instead of sets of numbers, relations or languages, as is usually done.

It turns out that (the polynomial-time versions of) most operators under consideration span the polynomial hierarchy. Thus, P = NP or, equivalently, the polynomial hierarchy collapses to P iff the class FP of all polytime functions is closed under one (or all) of these operators. This is shown in Section 7 after the introduction of fundamentals in Section 6. Now the nesting levels of the operators, starting with FP as the lowest level, are closely related to the stratification of the polynomial hierarchy, see Section 8. Valiant's sharp operator and the operator of summation are discussed from this point of view in Section 9. Section 10 shows that a collapse of the polynomial hierarchy can equivalently be expressed by closure properties of function classes under the polytime version of the first value operator. Moreover, this operator enables us to establish hierarchies between certain consecutive levels within the polynomial hierarchy which correspond to generalizations of the Boolean hierarchies over the classes Σ_k^p. These details are given in Section 11. The final Section 12 discusses the results and some open problems.

Throughout the paper, the reader is assumed to be familiar with basic notions, techniques and results of computability and complexity theory as they are dealt with in several textbooks cited in the references. Our terminology widely follows the recommendations by Soare [29,30]. So we speak of computable functions and decidable (or computable) sets as well as of c.e. (*i.e.*, computably enumerable) sets and relations. Notice, however, that computable functions always may be partial. If they are supposed to be total, this will explicitly be mentioned.

2. Function operators and their levels

Let FAll denote the class of all partial *arithmetical functions* $f : \mathbb{N}^m \rightarrowtail \mathbb{N}$ of arbitrary arities $m \geq 1$. By $f(\vec{x})\downarrow$, we indicate that $\vec{x} \in \mathrm{dom}(f)$, whereas $f(\vec{x})\uparrow$ or $f(\vec{x}) \simeq\uparrow$ means that $f(\vec{x})$ is undefined. Generally, $\simeq$ denotes the *conditional equality* between two values: both of them have to be simultaneously defined, or undefined, and must be equal in the first case.

A *function operator* or briefly *operator* is a partial mapping

$$\boldsymbol{\omega} : \bigcup_{n=1}^{\infty} \mathrm{FAll}^n \rightarrowtail \mathrm{FAll}.$$

The operator of *composition* is denoted by $\circ$. More precisely, we have $\circ(g, h_1, \ldots, h_l) = f$ if g is a function of some arity l, all the functions $h_1, \ldots, h_l$ and f have the same arity m, and $f(\vec{x}) \simeq g(h_1(\vec{x}), \ldots, h_l(\vec{x}))$ for all $\vec{x} \in \mathbb{N}^m$. We also write $g\circ(h_1, \ldots, h_l)$ instead of $\circ(g, h_1, \ldots, h_l)$, and simply $g\circ h_1$ if $l = 1$.

In many cases, the operators $\boldsymbol{\omega}$ themselves have fixed arities $n \geq 1$, *i.e.*, $\boldsymbol{\omega} : \mathrm{FAll}^n \rightarrowtail \mathrm{FAll}$. For example, the operator of *primitive recursion* is $\mathbf{pr} : \mathrm{FAll}^2 \rightarrowtail \mathrm{FAll}$, with $\mathbf{pr}(g, h) = f$ if $f(0, \vec{x}) \simeq g(\vec{x})$ and $f(y + 1, \vec{x}) \simeq h(y, \vec{x}, f(y, \vec{x}))$ for all $\vec{x} \in \mathbb{N}^m$, where g is of some arity $m \geq 1$ and both h and f have arity $m + 1$. Notice that here and in the sequel we simply write, *e.g.*, $(y, \vec{x})$ instead of $(y, x_1, \ldots, x_m)$, for $\vec{x} = (x_1, \ldots, x_m)$. The parameter-free primitive recursion, where a unary function $f : \mathbb{N} \rightarrowtail \mathbb{N}$ is defined by $f(0) = c \in \mathbb{N}$ and $f(y + 1) \simeq h(y, f(y))$ for some binary function h, can also be expressed by $\mathbf{pr}$, by means of composition, constant functions and projections. Indeed, let the projections $\varkappa_i^m : \mathbb{N}^m \longrightarrow \mathbb{N}$ yield the ith component of m-tupels, *i.e.*, $\varkappa_i^m(x_1, \ldots, x_m) = x_i$ for $m \geq 1, 1 \leq i \leq m$, and $\mathrm{const}_c(x) = c \in \mathbb{N}$ for all $x \in \mathbb{N}$. Then for the binary function $f' = \mathbf{pr}(\mathrm{const}_c, h\circ(\varkappa_1^3, \varkappa_3^3))$ the above defined function is obtained by $f = f'\circ(\varkappa_1^1, \varkappa_1^1)$.

The well known $\boldsymbol{\mu}$-operator of *minimalization* is unary, *i.e.*, it has arity 1. It assigns to any $(m + 1)$-ary function g an m-ary function $f = \boldsymbol{\mu}(g)$ defined by

$$f(\vec{x}) \simeq \begin{cases} y_0 & \text{if } g(y_0, \vec{x}) = 0, \text{ and } g(y, \vec{x})\downarrow \text{ and } g(y, \vec{x}) > 0 \text{ for all } y < y_0, \\ \uparrow & \text{if there is no } y_0 \text{ of the above kind.} \end{cases}$$

$\boldsymbol{\mu}$ must not be confused with the operator of *brutal minimalization*, $\overline{\boldsymbol{\mu}}$, which is defined by

$$\overline{\boldsymbol{\mu}}(g)(\vec{x}) \simeq \min\{y \,:\, g(y, \vec{x}) = 0\}, \quad \text{where } \min\emptyset \simeq\uparrow .$$

If g is a total function, then obviously $\boldsymbol{\mu}(g) = \overline{\boldsymbol{\mu}}(g)$. If $g(0, \vec{x})\uparrow$, it follows that $\boldsymbol{\mu}(g)(\vec{x})\uparrow$, whereas $\overline{\boldsymbol{\mu}}(g)(\vec{x})$ might still be defined however. The power of $\overline{\boldsymbol{\mu}}$ will be characterized in detail in the next section.

We shall see that $\overline{\boldsymbol{\mu}}$ is closely related to the *first value operator* $\boldsymbol{\phi}$. The latter assigns to any function g with arity, say, $m+1$ an m-ary function $f = \boldsymbol{\phi}(g)$ defined by

$$f(\vec{x}) \simeq \begin{cases} g(y_{\vec{x}}, \vec{x}) & \text{if there is an } y \in \mathbb{N} \text{ with } g(y, \vec{x}) \downarrow, \text{and } y_{\vec{x}} = \min\{y \in \mathbb{N}\colon\ g(y, \vec{x}) \downarrow\}, \\ \uparrow & \text{if } g(y, \vec{x}) \uparrow \text{ for all } y \in \mathbb{N}\,. \end{cases}$$

A more general version of $\boldsymbol{\phi}$ was used in [8] in order to characterize the classes of the Ershov hierarchy, *cf.* [9,12]. In [13], we introduced the denotation first value operator and used it in order to establish hierarchies of function classes. For a total function g, it always holds $\boldsymbol{\phi}(g)(\vec{x}) = g(0, \vec{x})$. So the operator $\boldsymbol{\phi}$ becomes only interesting if it is applied to properly partial functions.

The operator of inverting unary functions was known from the early days of computability theory, *cf.* [16,20], where it was applied only to surjective (and total) functions in order to yield total functions as results. Nowadays it has got considerable importance within structural complexity and cryptology, *cf.* [23,25], where it is usually restricted to injective functions however. Here we consider the operator $\boldsymbol{\varrho}$ which to any function $g : \mathbb{N} \rightarrowtail \mathbb{N}$ assigns a unary function $f = \boldsymbol{\varrho}(g)$, the *reverse* (also readable as *re*gular in*verse*) of g, defined by

$$f(y) \simeq \min\{x \ : \ g(x) = y\,\}, \qquad \text{where } \min \emptyset \simeq \uparrow.$$

By Kleene's Normal Form Theorem, every computable partial function f can be represented in the form $f = h \circ \boldsymbol{\mu}(g)$ with suitable primitive recursive functions h and g. On the other hand, the set FCom of all (partial) computable functions is closed under the operator $\boldsymbol{\mu}$. Let FPrim denote the set of all primitive recursive functions and FPaPrim the set of all *partial primitive recursive* functions. The latter ones are the restrictions of primitive recursive (total) functions to primitive recursive domains (*i.e.*, sets which are decidable by primitive recursive functions). Remember in this context that, at least in using the first value operator, it is essential to start with properly partial functions.

By $\mathrm{Clos}_{\{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots\}}(\,\mathrm{FC}\,)$ we denote the *closure* of a function class FC under the operators $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots$. This is the smallest class which includes FC and is closed under all $\boldsymbol{\omega}_i$, *i.e.*, it contains $\boldsymbol{\omega}_i(g_1, \ldots, g_n)$ whenever this is defined and the functions $g_1, \ldots, g_n$ belong to that class. Thus,

$$\mathrm{Clos}_{\{\circ, \boldsymbol{\mu}\}}(\,\mathrm{FPrim}\,) = \mathrm{Clos}_{\{\circ, \boldsymbol{\mu}\}}(\,\mathrm{FPaPrim}\,) = \mathrm{FCom} = \mathrm{Clos}_{\{\circ, \boldsymbol{\mu}\}}(\,\mathrm{FCom}\,).$$

Our first goal is the characterization of the sets $\mathrm{Clos}_{\{\circ, \boldsymbol{\omega}\}}(\,\mathrm{FC}\,)$ for $\boldsymbol{\omega} \in \{\overline{\boldsymbol{\mu}}, \boldsymbol{\phi}, \boldsymbol{\varrho}\}$ and $\mathrm{FC} \in \{\mathrm{FPrim}, \mathrm{FPaPrim}, \mathrm{FCom}\}$. To get more detailed results, we consider the nesting degrees of functions with respect to the operators $\boldsymbol{\omega}$. This technique goes back to the early sixties of the past century when degrees of primitive recursive functions with respect to the operator **pr** were studied. Related results and references can be found, *e.g.*, in [3,19,22,24]. In order to avoid confusions with the Turing, or other degrees, throughout this paper we shall denote nesting degrees (with respect to arbitrary operators $\boldsymbol{\omega}$) as ($\boldsymbol{\omega}$-)levels.

For a function class FC and an operator $\boldsymbol{\omega}$, let $\boldsymbol{\omega}(\mathrm{FC})$ denote the set of all functions obtained by applying the operator $\boldsymbol{\omega}$ exactly once to arguments from FC. In particular, if $\boldsymbol{\omega}$ is unary as usual in the cases we shall mainly deal with, $\boldsymbol{\omega}(\mathrm{FC})$ is just the image set of FC under $\boldsymbol{\omega}$. The $\boldsymbol{\omega}$*-levels* are the following function classes $\mathrm{FLev}_{\boldsymbol{\omega}}(k)$, for all natural numbers k:

$$\begin{aligned} \mathrm{FLev}_{\boldsymbol{\omega}}(0) &= \mathrm{FPaPrim} \quad \text{and} \\ \mathrm{FLev}_{\boldsymbol{\omega}}(k+1) &= \mathrm{Clos}_{\{\circ\}}\left(\, \mathrm{FLev}_{\boldsymbol{\omega}}(k) \cup \boldsymbol{\omega}(\mathrm{FLev}_{\boldsymbol{\omega}}(k))\,\right) \quad \text{for all } k \in \mathbb{N}. \end{aligned}$$

This means that $\mathrm{FLev}_{\boldsymbol{\omega}}(k)$ contains just the functions obtained from $\mathrm{FLev}_{\boldsymbol{\omega}}(0) =$ FPaPrim by applying the composition and the operator $\boldsymbol{\omega}$, where applications of $\boldsymbol{\omega}$ are nested at most k times. Immediately, this would yield $\mathrm{FLev}_{\boldsymbol{\omega}}(k+1) = \mathrm{Clos}_{\{\circ\}}(\,\mathrm{FLev}_{\boldsymbol{\omega}}(k) \cup \boldsymbol{\omega}(\mathrm{Clos}_{\{\circ\}}(\,\mathrm{FLev}_{\boldsymbol{\omega}}(k)\,))\,)$. Since all the levels $\mathrm{FLev}_{\boldsymbol{\omega}}(k)$ are closed under composition, however, the original form can be simplified to the above one. Obviously, we have $\mathrm{Clos}_{\{\circ,\boldsymbol{\omega}\}}(\,\mathrm{FPaPrim}\,) = \bigcup_{k=1}^{\infty} \mathrm{FLev}_{\boldsymbol{\omega}}(k)$ and $\mathrm{FLev}_{\boldsymbol{\omega}}(k) \subseteq \mathrm{FLev}_{\boldsymbol{\omega}}(k+1)$.

By Kleene's Normal Form Theorem, all nonzero $\boldsymbol{\mu}$-levels coincide:

$$\mathrm{FLev}_{\boldsymbol{\mu}}(k) = \mathrm{FCom} \quad \text{for all } k \geq 1.$$

In the next section we shall see that $\overline{\boldsymbol{\mu}}$, $\boldsymbol{\phi}$ and $\boldsymbol{\varrho}$ yield more interesting levels. Now we first show that these three operators lead to the same hierarchy of levels.

Proposition 2.1. *For any* $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \in \{\overline{\boldsymbol{\mu}}, \boldsymbol{\phi}, \boldsymbol{\varrho}\}$ *and* $k \in \mathbb{N}$, *we have* $\mathrm{FLev}_{\boldsymbol{\omega}_1}(k) = \mathrm{FLev}_{\boldsymbol{\omega}_2}(k)$.

This follows by induction on k. The case $k = 0$ is trivial. To prove that under the premise $\mathrm{FLev}_{\boldsymbol{\omega}_1}(k) = \mathrm{FLev}_{\boldsymbol{\omega}_2}(k)$ it follows $\mathrm{FLev}_{\boldsymbol{\omega}_1}(k+1) \subseteq \mathrm{FLev}_{\boldsymbol{\omega}_2}(k+1)$, it is enough to show the inclusion $\boldsymbol{\omega}_1(\mathrm{FLev}_{\boldsymbol{\omega}_1}(k)) \subseteq \mathrm{Clos}_{\{\circ\}}(\mathrm{FLev}_{\boldsymbol{\omega}_2}(k) \cup \boldsymbol{\omega}_2(\mathrm{FLev}_{\boldsymbol{\omega}_2}(k)))$, where the second class can be supposed to be equal to $\mathrm{Clos}_{\{\circ\}}(\,\mathrm{FLev}_{\boldsymbol{\omega}_1}(k) \cup \boldsymbol{\omega}_2(\mathrm{FLev}_{\boldsymbol{\omega}_1}(k))\,)$. So the proof of Proposition 2.1 is completed by showing the following lemma.

Lemma 2.1. *For* $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \in \{\overline{\boldsymbol{\mu}}, \boldsymbol{\phi}, \boldsymbol{\varrho}\}$ *and every function class* FC *which is closed under composition and satisfies* $\mathrm{FPaPrim} \subseteq \mathrm{FC} \subseteq \mathrm{FAll}$, *we have*

$$\boldsymbol{\omega}_1(\mathrm{FC}) \subseteq \mathrm{Clos}_{\{\circ\}}\left(\, \mathrm{FC} \cup \boldsymbol{\omega}_2(\mathrm{FC})\,\right).$$

This inclusion says that the operator $\boldsymbol{\omega}_1$ on FC can be expressed by $\circ$ and $\boldsymbol{\omega}_2$, where no nesting of $\boldsymbol{\omega}_2$ is needed. Only some partial primitive recursive functions have to be additionally employed. In fact, we need for this only very special functions. We shall return to this point in Section 6.

The proof of the lemma employs the m-tupling functions $\tau^m : \mathbb{N}^m \longrightarrow \mathbb{N}$. More precisely, let $\tau^m(x_1, \ldots, x_m)$ or $\langle x_1, \ldots, x_m \rangle$ denote the *Cantor number* of the m-tuple $\vec{x} = (x_1, \ldots, x_m)$. For $m \geq 2$, the functions τ^m and their inverses $\pi_i^m : \mathbb{N} \longrightarrow \mathbb{N}$, $1 \leq i \leq m$, are obtained from Cantor's pairing function τ^2 and its inverses π_1^2 and π_2^2 by inductive definition over m. $\tau^1 = \pi_1^1$ is the identity on $\mathbb{N}$. We have always

$$\tau^m(\pi_1^m(x), \ldots, \pi_m^m(x)) = x \quad \text{and} \quad \pi_i^m \circ \tau^m(x_1, \ldots, x_m) = x_i \ \ (1 \le i \le m).$$

Both τ^m and all π_i^m are primitive recursive, τ^m is a bijection of $\mathbb{N}^m$ onto $\mathbb{N}$, the functions π_i^m are injective. Moreover, any τ^m is monotonous with respect to each argument, *i.e.*,

$$x_i < x_i' \text{ iff } \tau^m(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_m) < \tau^m(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_m).$$

Now we show that $\boldsymbol{\mu}$, $\boldsymbol{\phi}$ and $\boldsymbol{\varrho}$ can be expressed by each other in the above mentioned sense. More precisely, a cycle of three such expressibilities will be given.

(i) Expressing $\overline{\boldsymbol{\mu}}$ by $\boldsymbol{\phi}$: For $f = \overline{\boldsymbol{\mu}}(g)$ with $g \in \mathrm{FC}$, we put

$$g'(y, \vec{x}) \simeq \begin{cases} y & \text{if } g(y, \vec{x}) = 0, \\ \uparrow & \text{otherwise.} \end{cases}$$

Then $f = \boldsymbol{\phi}(g')$. Moreover, the function h defined by

$$h(y, z) \simeq \begin{cases} z & \text{if } y = 0, \\ \uparrow & \text{otherwise,} \end{cases}$$

is partial primitive recursive. If f is m-ary, it holds $g'(y, \vec{x}) \simeq h(g(y, \vec{x}), \varkappa_1^{m+1}(y, \vec{x}))$, hence $g' = h \circ (g, \varkappa_1^{m+1}) \in \mathrm{FC}$.

(ii) Expressing $\boldsymbol{\phi}$ by $\boldsymbol{\varrho}$: For an m-ary function $f = \boldsymbol{\phi}(g)$ with $g \in \mathrm{FC}$, let

$$g'(\langle y, \vec{x} \rangle) \simeq \begin{cases} \langle \vec{x} \rangle & \text{if } g(y, \vec{x}){\downarrow}, \\ \uparrow & \text{otherwise.} \end{cases}$$

Then $g' = h \circ (g, \tau^m \circ (\pi_2^{m+1}, \ldots, \pi_{m+1}^{m+1})) \in \mathrm{FC}$ with a suitable function $h \in \mathrm{FPaPrim}$. Moreover, we have $f(\vec{x}) \simeq g(\pi_1^{m+1} \circ \boldsymbol{\varrho}(g')(\langle \vec{x} \rangle), \vec{x})$, *i.e.*, $f = g \circ (\pi_1^{m+1} \circ \boldsymbol{\varrho}(g') \circ \tau^m, \pi_1^m, \ldots, \pi_m^m)$.

(iii) Expressing $\boldsymbol{\varrho}$ by $\overline{\boldsymbol{\mu}}$: If $f = \boldsymbol{\varrho}(g)$ for (unary) functions f and g with $g \in \mathrm{FC}$, we consider the binary function

$$g'(y, z) \simeq \begin{cases} 0 & \text{if } g(y) = z, \\ \uparrow & \text{otherwise.} \end{cases}$$

It is easily seen that $g' \in \mathrm{Clos}_{\{\circ\}}(\mathrm{FPaPrim} \cup \{g\}) \subseteq \mathrm{FC}$ and $f(z) \simeq \overline{\boldsymbol{\mu}}(g')(z)$, *i.e.*, $f = \overline{\boldsymbol{\mu}}(g')$.

This completes the proof of Lemma 2.1 and Proposition 2.1. □

In the sequel, we shall simply write $\mathrm{FLev}(k)$ instead of $\mathrm{FLev}_{\boldsymbol{\omega}}(k)$ with $\boldsymbol{\omega} \in \{\overline{\boldsymbol{\mu}}, \boldsymbol{\phi}, \boldsymbol{\varrho}\}$.

3. Relationships to the arithmetical hierarchy

In order to describe the precise meanings of notations used in the sequel, we first have to recall some basic facts and notions of computability theory. Most of them are folklore, more details can be found in [6,17,21,28,30,33].

The *arithmetical hierarchy* is usually considered both as the hierarchy of the classes Σ_k, Π_k and Δ_k, for $k \in \mathbb{N}$, but also as the union over all these classes: $\mathrm{AH} = \bigcup_{k=0}^{\infty} \Sigma_k$. Here Σ_k contains just those relations $A \subseteq \mathbb{N}^m$, $m \geq 1$, which are representable in the form

$$A = \{\vec{x} \in \mathbb{N}^m : \exists y_1 \forall y_2 \dots Q y_k (y_1, \dots, y_k, \vec{x}) \in B\}$$

with $B \in \mathrm{Com}$, which is the class of all *computable relations*, and $Q \in \{\exists, \forall\}$, so that the prefix of quantifiers in the above representation becomes alternating. A relation $A \subseteq \mathbb{N}^m$ belongs to AH iff it is representable in the first-order logic of the structure $\langle \mathbb{N}; 0, 1; \leq; +, \cdot \rangle$, the so-called *elementary arithmetic*. The classes of the arithmetical hierarchy can also be characterized by means of relative computability and the jump operator, as will be sketched now.

We start with a standard numbering $(\mathcal{M}_n : n \in \mathbb{N})$ of all *oracle Turing machines (OTMs)*. Notice that the syntax of a machine $\mathcal{M}_n$, and hence its index or encoding n, does not depend on the oracle set. By $\mathcal{M}_n^A$, we indicate that the machine $\mathcal{M}_n$ is working with the oracle set $A \subseteq \mathbb{N}$, whereas $\mathcal{M}_n^A(\vec{x})$ means the (finite or infinite) computation of $\mathcal{M}_n^A$ on the input $\tau^m(\vec{x}) = \langle \vec{x} \rangle$. Let Φ_n^A denote the unary arithmetical function computed by $\mathcal{M}_n^A$. Thus, $(\Phi_n^A : n \in \mathbb{N})$ is a numbering of all unary functions computable *in* (*i.e.*, relatively to) A, and $(\Phi_n^A \circ \tau^m : n \in \mathbb{N})$ is a standard numbering of the m-ary functions computable in A. The numberings with the empty oracle, $(\Phi_n^\emptyset : n \in \mathbb{N})$ and $(\Phi_n^\emptyset \circ \tau^m : n \in \mathbb{N})$, respectively, correspond to absolute computability.

The *jump* of a set $A \subseteq \mathbb{N}$ is defined by $A' = \{n : \Phi_n^A(n) \downarrow\}$. The sequence $(\emptyset^{(k)} : k \in \mathbb{N})$, where $\emptyset^{(0)} = \emptyset$ and $\emptyset^{(k+1)} = {\emptyset^{(k)}}'$, spans the arithmetical hierarchy in the following sense.

Fact 3.1. *For all $k \geq 1$, $\emptyset^{(k)}$ is Σ_k-complete (with respect to* m*-reducibility $\leq_{\mathrm{m}}$), and a relation $A \in \mathbb{N}^m$ belongs to Σ_k iff it is computably enumerable (c.e.) in $\emptyset^{(k-1)}$ or, equivalently, in some $B \in \Sigma_{k-1}$.*

A function $f \in \mathrm{FAll}$ is said to be *arithmetically representable* iff its *graph*,

$$\mathrm{graph}(f) = \{(\vec{x}, y) : f(\vec{x}) = y\},$$

belongs to AH. Let FAH denote the set of all arithmetically representable functions. It is quite natural to transfer the stratification of the arithmetical hierarchy to the function class FAH by considering the classes

$$\mathrm{F}\Sigma_k = \{f : \mathrm{graph}(f) \in \Sigma_k\}, \text{ for } k \in \mathbb{N}.$$

In particular, $F\Sigma_1 = \text{FCom}$. The special class $F\Sigma_0$ is less interesting in the present context. Since a function f is computable in a set A iff graph(f) is c.e. in A, we have

Fact 3.2. *For $k \geq 1$ and functions f, $f \in F\Sigma_k$ iff f is computable in $\emptyset^{(k-1)}$ or, equivalently, in some $B \in \Sigma_{k-1}$, i.e., $f = \Phi_n^B \circ \tau^m$ for some $n \in \mathbb{N}$, if f is m-ary.*

For $A \subseteq \mathbb{N}^m$, the relation $A \times \{0\}$ is the graph of the *semicharacteristic function* χ_A^0 of A, which is defined by

$$\chi_A^0(\vec{x}) \simeq \begin{cases} 0 & \text{if } \vec{x} \in A, \\ \uparrow & \text{otherwise.} \end{cases}$$

Since A and $A \times \{0\}$ are computably isomorphic, we have $A \in \Sigma_k$ iff $\chi_A^0 \in F\Sigma_k$. This shows how the classes Σ_k can be defined by means of the $F\Sigma_k$. Moreover, $\chi^0_{\emptyset^{(k)}} \in F\Sigma_k \setminus F\Sigma_{k-1}$ for any $k \geq 1$.

Since the arithmetical representability of functions is hereditary under $\circ$ and $\boldsymbol{\omega} \in \{\overline{\boldsymbol{\mu}}, \boldsymbol{\phi}, \boldsymbol{\varrho}\}$, one easily sees that $\text{Clos}_{\{\circ,\boldsymbol{\omega}\}}(\text{FPaPrim}) \subseteq \text{FAH}$. Now we shall prove not only the converse inclusion but also that the nonzero levels of functions are closely related to the stratification of the arithmetical hierarchy.

Proposition 3.1. *For all $k \in \mathbb{N}$, $\text{FLev}(1 + 2k) = F\Sigma_{1+k}$.*

The proof is by induction on k. For $k = 0$, we have to show that $\text{FLev}(1) = \text{FCom}$. The inclusion "$\supseteq$" holds by Kleene's Normal Form Theorem and by the fact that $\boldsymbol{\mu}(f) = \overline{\boldsymbol{\mu}}(f)$ for any (total) primitive recursive function f. "$\subseteq$" follows since $\boldsymbol{\phi}(\text{FPaPrim})$ as well as FPaPrim are subclasses of FCom, and this is closed under composition.

Now we suppose $\text{FLev}(1 + 2k) = F\Sigma_{1+k}$ and conclude

$$\text{FLev}(1 + 2k + 2) = F\Sigma_{1+k+1}.$$

To show the inclusion "$\subseteq$", let $g \in F\Sigma_{1+k}$, *i.e.*, g be computable in $\emptyset^{(k)}$. If $\overline{\boldsymbol{\mu}}(g)$ is defined (*i.e.*, g is of an arity $m \geq 2$), then $\overline{\boldsymbol{\mu}}(g)$ is computable in $\emptyset^{(k+1)}$ and $\text{dom}(\overline{\boldsymbol{\mu}}(g))$ is even decidable (computable) in $\emptyset^{(k+1)}$. The latter follows since $\vec{x} \in \text{dom}(\overline{\boldsymbol{\mu}}(g))$ iff there is a number y with $g(y, \vec{x}) = 0$, and this property is c.e. in $\emptyset^{(k)}$, hence it is decidable in $\emptyset^{(k+1)}$. Since the decidability of the domains is hereditary under composition of functions, all functions $f \in \text{FLev}(1 + 2k + 1) = \text{Clos}_{\{\circ\}}(F\Sigma_{1+k} \cup \overline{\boldsymbol{\mu}}(F\Sigma_{1+k}))$ have domains decidable in $\emptyset^{(k+1)}$. This shows already that they build a proper subclass of $F\Sigma_{1+k+1}$. Moreover, for all such functions of arities $\geq$2, it follows that $\overline{\boldsymbol{\mu}}(f)$ is computable in $\emptyset^{(k+1)}$ too. Hence we have $\text{FLev}(1 + 2k + 2) = \text{Clos}_{\{\circ\}}(\text{FLev}(1 + 2k + 1) \cup \overline{\boldsymbol{\mu}}(\text{FLev}(1 + 2k + 1))) \subseteq F\Sigma_{1+k+1}$.

To show the converse inclusion "$\supseteq$", let $f \in F\Sigma_{1+k+1}$, *i.e.*, $f = \Phi_{n_f}^{\emptyset^{(k+1)}}$ for some $n_f \in \mathbb{N}$. The following proof of $f \in \text{FLev}(1 + 2k + 2)$ again employs standard techniques of computability theory. For reasons of readability, we prefer an informal description.

The finite (halting) computations $\mathcal{M}_{n_f}^{\emptyset^{(k+1)}}(\vec{x})$, considered as sequences of configurations of the OTM $\mathcal{M}_{n_f}$, can be encoded by natural numbers n_c. Also, the finite sequences of oracle queries, which are positively or negatively answered in the course of a finite computation, can be encoded by numbers n_+ and n_-, respectively. All this can be done in some standard way (sometimes called Gödelization) such that there are primitive recursive functions h_1 and h_2 satisfying

$h_1(\vec{x}, n_c, n_+, n_-) = 0$ iff n_c encodes a halting computation of $\mathcal{M}_{n_f}$ starting with input $\langle\vec{x}\rangle$, in the course of which exactly the oracle queries encoded by n_+ are positively answered and those encoded by n_- are negatively answered

and

$y = h_2(n_c)$ is always the output produced by the computation encoded by n_c.

Moreover, we use two functions, g_+ and g_-, of higher levels such that

$$g_+(n_+) = 0 \quad \text{iff} \quad \text{all queries encoded by } n_+ \text{ belong to } \emptyset^{(k+1)}$$

and

$$g_-(n_+) = 0 \quad \text{iff} \quad \text{all queries encoded by } n_- \text{ do not belong to } \emptyset^{(k+1)}.$$

Then we have

$$f(\vec{x}) = y \quad \text{iff} \quad \exists\langle n_c, n_+, n_-\rangle(\, h_1(\vec{x}, n_c, n_+, n_-) = 0 \wedge \\ y = h_2(n_c) \wedge g_+(n_+) = 0 \wedge g_-(n_-) = 0\,),$$

hence

$$f(\vec{x}) \simeq h_2 \circ \varkappa_1^3 \circ \overline{\boldsymbol{\mu}}(h)(\vec{x}),$$

where $h(\langle n_c, n_+, n_-\rangle, \vec{x}) \simeq h_1(\vec{x}, n_c, n_+, n_-) + g_+(n_+) + g_-(n_-)$. So the proof is completed by showing that $g_+, g_- \in \mathrm{FLev}(1+2k+1) = \mathrm{Clos}_{\{\circ\}}(\,\mathrm{F}\Sigma_{1+k} \cup \overline{\boldsymbol{\mu}}(\mathrm{F}\Sigma_{1+k}))$.

$g_+(n_+) = 0$ has to confirm that the finitely many queries q encoded by n_+ belong to $\emptyset^{(k+1)} = \{n : \Phi_n^{\emptyset^{(k)}}(n)\downarrow\}$. To compute $g_+(n_+)$, all such computations $\mathcal{M}_q^{\emptyset^{(k)}}(q)$ are tried to simulate. If $q \in \emptyset^{(k+1)}$ for all queries q encoded by n_+, put $g_+(n_+) = 0$, otherwise let $g_+(n_+)\uparrow$. Thus, g_+ can be computed by means of the oracle set $\emptyset^{(k)}$, hence $g_+ \in \mathrm{F}\Sigma_{1+k}$.

Similarly, one gets a function $g_0 \in \mathrm{F}\Sigma_{1+k}$ satisfying

$$g_0(0, n_-) \simeq \begin{cases} 0 & \text{if there is a query } q \text{ encoded by } n_- \text{ with } q \in \emptyset^{(k+1)}, \\ \uparrow & \text{otherwise,} \end{cases}$$

$$g_0(1, n_-) = 0 \quad \text{for all } n_- \in \mathbb{N}.$$

Then we have

$$\overline{\boldsymbol{\mu}}(g_0)(n_-) = \begin{cases} 1 & \text{if } q \notin \emptyset^{(k+1)} \text{ for all queries } q \text{ encoded by } n_-, \\ 0 & \text{otherwise,} \end{cases} .$$

Thus, $g_-(n_-) \simeq 1 - \overline{\boldsymbol{\mu}}(g_0)(n_-)$ defines a function $g_- \in \text{FLev}(1 + 2k + 1)$ with all the properties we required. □

In the light of Propositions 2.1 and 3.1, it becomes interesting to explore how some further operators known from computability theory and complexity theory are related to each other and to the arithmetical hierarchy.

Of course, for operators $\boldsymbol{\omega}$ which always yield computable functions if they are applied to computable ones, it holds $\text{Clos}_{\{\circ,\boldsymbol{\omega}\}}(\text{FPaPrim}) \subseteq \text{FCom}$. Hence they cannot span the arithmetical hierarchy of functions. A first such example is the classical operator $\boldsymbol{\mu}$. Some further ones are given by operators $\boldsymbol{\omega}_f$ defined by computable functions f (of arity, say, m) according to $\boldsymbol{\omega}_f(g_1, \ldots, g_m) = f \circ (g_1, \ldots, g_m)$.

4. The limit operator and the arithmetical hierarchy

The *limit operator* is known to be rather powerful. It assigns to each total $(m+1)$-ary function g the m-ary function $f = \mathbf{lim}(g)$ defined by

$$f(\vec{x}) \simeq \begin{cases} z & \text{if there is a } y_0 \in \mathbb{N} \text{ such that } g(y, \vec{x}) = z \text{ for all } y \geq y_0, \\ \uparrow & \text{if there are no such } z \text{ and } y_0. \end{cases}$$

Obviously, the arithmetical representability of functions is hereditary under the limit operator. From this it follows inductively that $\text{FLev}_{\mathbf{lim}}(k) \subseteq \text{FAH}$ for all $k \in \mathbb{N}$, and we have $\text{Clos}_{\{\circ,\mathbf{lim}\}}(\text{FPaPrim}) \subseteq \text{FAH}$. On the other hand, it is easily seen that $\text{FCom} \subseteq \mathbf{lim}(\text{FPrim})$, hence $\text{Clos}_{\{\mathbf{lim}\}}(\text{FCom}) = \text{Clos}_{\{\mathbf{lim}\}}(\text{FPrim})$. By the generalized version of Shoenfield's Limit Lemma, *cf.* Proposition IV.1.19 in [17], it follows even that $\text{FAH} \subseteq \text{Clos}_{\{\mathbf{lim}\}}(\text{FCom})$, thus $\text{FAH} \subseteq \text{Clos}_{\{\circ,\mathbf{lim}\}}(\text{FPaPrim})$. So we can conclude that $\text{Clos}_{\{\circ,\mathbf{lim}\}}(\text{FPaPrim}) = \text{FAH}$.

Nevertheless, with respect to the nesting levels the limit operator is more powerful than $\overline{\boldsymbol{\mu}}$, $\boldsymbol{\phi}$ and $\boldsymbol{\varrho}$. For example, the (total) characteristic function of $\emptyset'$, $\chi_{\emptyset'}$, belongs to $\text{FLev}_{\mathbf{lim}}(1)$ (but it does not belong to $\text{F}\Sigma_1 = \text{FLev}(1)$ as one knows). To show this, we employ the functions $\Phi^A_{n|s}$ obtained by simulating at most s steps of the machines $\mathcal{M}_n$ with oracle A. More precisely, let

$$\Phi^A_{n|s}(\vec{x}) \simeq \begin{cases} z & \text{if } \mathcal{M}^A_n(\vec{x}) \text{ halts after at most } s \text{ steps and yields the output } z, \\ \uparrow & \text{otherwise.} \end{cases}$$

For any $A \subseteq \mathbb{N}$, the function $g(s, n, \vec{x}) \simeq \Phi^A_{n|s}(\vec{x})$ is computable in A, and its domain is decidable in A; for $A = \emptyset$, we even have primitive recursivity in both cases. Now we can put

$$h(s,n) = \begin{cases} \Phi^{\emptyset}_{n|s}(n) & \text{if } \Phi^{\emptyset}_{n|s}(n)\downarrow, \\ 0 & \text{otherwise.} \end{cases}$$

Then $h \in \mathrm{FPrim} \subseteq \mathrm{FLev}(0)$, and we have $\chi_{\emptyset'} = \mathbf{lim}(h)$.

Even if $\mathrm{FLev}_{\mathbf{lim}}(1) \neq \mathrm{F}\Sigma_1$, there is a close correspondence between the levels of the limit operator and the stratification of the arithmetical hierarchy of functions.

Proposition 4.1. *For all* $k \geq 1$, $\mathrm{FLev}_{\mathbf{lim}}(k) = \mathrm{F}\Sigma_{1+k}$.

The proof is by induction on k. We first remark that $\mathrm{F}\Sigma_2$, as any $\mathrm{F}\Sigma_k$, is closed under composition. Thus, $\mathrm{FLev}_{\mathbf{lim}}(1) \subseteq \mathrm{F}\Sigma_2$ follows from $\mathrm{FPaPrim} \cup \mathbf{lim}(\mathrm{FPaPrim}) \subseteq \mathrm{F}\Sigma_2$. Of course, $\mathrm{FPaPrim} \subseteq \mathrm{F}\Sigma_2$ holds trivially. Let $f = \mathbf{lim}(g)$ for some $g \in \mathrm{FPaPrim}$, *i.e.*, $g \in \mathrm{FPrim}$, since the limit operator is only defined on total functions. Thus,

$$(\vec{x}, z) \in \mathrm{graph}(f) \quad \text{iff} \quad \exists y_0 \forall y (y \geq y_0 \rightarrow g(y, \vec{x}) = z).$$

This shows that $\mathrm{graph}(f) \in \Sigma_2$, hence $f \in \mathrm{F}\Sigma_2$.

Now let $f \in \mathrm{F}\Sigma_2$ be an m-ary function. Then f is computable in $\emptyset'$, *i.e.*, $f = \Phi^{\emptyset'}_{n_f} \circ \tau^m$ for some $n_f \in \mathbb{N}$. Instead of $\emptyset' = \{n : \Phi^{\emptyset}_n(n)\downarrow\}$, the following approximations A'_y are used:

$$A'_y = \{n : \mathcal{M}^{\emptyset}_n(n) \text{ halts after at most } y \text{ steps}\}.$$

Obviously, $A'_y \subseteq A'_{y+1}$ and $\emptyset' = \bigcup_{y \in \mathbb{N}} A'_y$. For $y \in \mathbb{N}$ and $\vec{x} \in \mathbb{N}^m$, let

$$g(\langle y,z\rangle, \vec{x}) = \begin{cases} \Phi^{A'_z}_{n_f|y}(\vec{x}) & \text{if } \Phi^{A'_z}_{n_f|y}(\vec{x})\downarrow \text{ and only oracle queries to values } q \leq y \\ & \quad \text{are put in the course of the related computation,} \\ y & \text{if } \Phi^{A'_z}_{n_f|y}(\vec{x})\uparrow \text{ or an oracle query to a value } q > y \\ & \quad \text{is put within the first } y \text{ steps of } \mathcal{M}^{A'_z}_{n_f}(\vec{x}). \end{cases}$$

In this way, a (total) $(m{+}1)$-ary function g is defined. The first argument is written as a Cantor number in order to simplify the description. By standard arguments of computability theory, it follows that g is even primitive recursive. In particular, $g(\langle y,z\rangle, \vec{x})$ can be computed by simulating at most y steps of the computation $\mathcal{M}^{A'_z}_{n_f}(\vec{x})$, as long as all oracle queries concern values $q \leq y$. Moreover, the queries "$q \in A'_z$?" can be decided by simulating at most z steps of the computations $\mathcal{M}^{\emptyset}_q(q)$.

If $\mathcal{M}^{\emptyset'}_{n_f}(\vec{x})$ halts at all, then, for sufficiently large y, it halts after $\leq y$ steps and puts only oracle queries to values $q \leq y$. Hence for sufficiently large z it holds $q \in \emptyset'$ iff $q \in A'_z$. Thus, $g(\langle y,z\rangle, \vec{x}) = f(\vec{x})$ for $y \geq y_0$ and $z \geq z_0$ with suitably chosen numbers y_0 and z_0. This means $\mathbf{lim}(g)(\vec{x}) = f(\vec{x})$.

If $\Phi^{\emptyset'}_{n_f}(\vec{x})\uparrow$, then the computation $\mathcal{M}^{\emptyset'}_{n_f}(\vec{x})$ never halts. Then to every y there is a sufficiently large z such that for all oracle queries q, which are put up to the step y of computation, it holds $q \in \emptyset'$ iff $q \in A'_z$. Thus, the computation $\mathcal{M}^{A'_z}_{n_f}(\vec{x})$ performs at least $y+1$ steps, *i.e.*, $g(\langle y,z\rangle,\vec{x}) = y$. It follows that $\mathbf{lim}(g)(\vec{x})\uparrow$, and we have shown that $\mathrm{F}\Sigma_2 = \mathbf{lim}(\mathrm{FPrim})$.

Now we show that from $\mathrm{F}\Sigma_{k+1} = \underbrace{\mathbf{lim}(\ldots\mathbf{lim}}_{k\text{ times}}(\mathrm{FPrim})\ldots)$, for some $k \geq 1$, it follows

$$\mathrm{F}\Sigma_{k+2} = \underbrace{\mathbf{lim}(\ldots\mathbf{lim}}_{k+1\text{ times}}(\mathrm{FPrim})\ldots).$$

The inclusion from left to right can be proved as above for the special case $k = 1$: Supposed $f \in \mathrm{F}\Sigma_{k+2}$, hence $f = \Phi^{\emptyset^{(k+1)}}_{n_f} \circ \tau^m$ for some $n_f \in \mathbb{N}$, we put

$$A^{(k+1)}_y = \{n : \mathcal{M}^{\emptyset^{(k)}}_n(n) \text{ halts after at most } y \text{ steps}\}$$

and

$$g(\langle y,z\rangle,\vec{x}) = \begin{cases} \Phi^{A^{(k+1)}_z}_{n_f|y}(\vec{x}) & \text{if } \Phi^{A^{(k+1)}_z}_{n_f|y}(\vec{x})\downarrow \text{ and only oracle queries to values } q \leq y \\ & \quad \text{are put in the course of the related computation,} \\ y & \text{if } \Phi^{A^{(k+1)}_z}_{n_f|y}(\vec{x})\uparrow \text{ or an oracle query to a value } q > y \\ & \quad \text{is put within the first } y \text{ steps of } \mathcal{M}^{A^{(k+1)}_z}_{n_f}(\vec{x}). \end{cases}$$

The total function g is computable in $\emptyset^{(k)}$, *i.e.* $g \in \mathrm{F}\Sigma_{k+1}$, and $f = \mathbf{lim}(g)$.

To conclude the converse inclusion, and even $\mathrm{FLev}_{\mathbf{lim}}(k+1) \subseteq \mathrm{F}\Sigma_{k+2}$ from the inductive hypothesis of the proof of Proposition 4.1, it is enough to show that $\mathbf{lim}(\mathrm{F}\Sigma_{k+1}) \subseteq \mathrm{F}\Sigma_{k+2}$. So let $f = \mathbf{lim}(g)$ for some $g \in \mathrm{F}\Sigma_{k+1}$, *i.e.*, g is computable in $\emptyset^{(k)}$. Then the function g' defined by

$$g'(y,\vec{x}) \simeq \begin{cases} 0 & \text{if there is a } y' \geq y \text{ such that } g(y',\vec{x}) \neq g(y,\vec{x}), \\ \uparrow & \text{otherwise } (i.e., g(y',\vec{x}) = g(y,\vec{x}) \text{ for all } y' > y), \end{cases}$$

is computable in $\emptyset^{(k)}$ too, hence $g' \in \mathrm{F}\Sigma_{k+1}$. Thus, $\mathrm{dom}(g')$ is decidable (computable) in $\emptyset^{(k+1)}$. By means of the computation of g and the decision of $\mathrm{dom}(g')$, however, $\mathbf{lim}(g)$ can be shown to be computable in $\emptyset^{(k+1)}$. This means $f \in \mathrm{F}\Sigma_{k+2}$, and the inductive proof is complete. □

By the above proof, we have the following normal form representation of the arithmetically representable functions with respect to the limit operator. It can be seen as a variant of the generalized limit lemma which usually concerns the limit representations of characteristic functions of sets from Δ_k in the classical arithmetical hierarchy.

Corollary 4.1. *For $k \geq 1$, an m-ary function f belongs to $\mathrm{F}\Sigma_{k+1}$ iff there is a $(k+m)$-ary function $g \in \mathrm{FPrim}$ such that $f = \mathbf{lim}^k(g)$.* □

Herein $\mathbf{lim}^k(g)$ means the result of the k-fold iterated limit operator applied to g, where it is supposed that $\mathbf{lim}^{k'}(g)$ is a total function for $1 \leq k' < k$.

It might be of interest that the limit operator could also be understood in the generalized way that

$$\mathbf{lim}(g)(\vec{x}) \simeq \begin{cases} z & \text{if } g(y,\vec{x}) = z \text{ for almost all } y \in \mathbb{N}, \\ \uparrow & \text{if there is no such } z, \end{cases}$$

for every (*i.e.*, not necessarily total) function g of an arity ≥ 2. Also with this meaning, Proposition 4.1 and Corollary 4.1 remain true, as the above proof shows. Nevertheless, we continue to consider the limit operator as applicable to total functions only.

5. Some further operators

Next we consider three further operators, *maximum*, *sharp* and *summation*, each of which defines the same hierarchy of levels as the limit operator. They all assign to arbitrary $(m+1)$-ary functions g certain m-ary functions $\mathbf{max}(g)$, $\sharp(g)$ and $\mathbf{sum}(g)$, respectively, which are defined by

$$\begin{aligned}
\mathbf{max}(g)(\vec{x}) &\simeq \begin{cases} \max\{g(y,\vec{x}) : g(y,\vec{x})\!\downarrow\} & \text{if this set is nonempty and finite,} \\ \uparrow & \text{if the above set is empty or infinite,} \end{cases} \\
\sharp(g)(\vec{x}) &\simeq \begin{cases} \mathrm{card}\{y : g(y,\vec{x})\!\downarrow\} & \text{if this set is finite,} \\ \uparrow & \text{if the above set is infinite,} \end{cases} \\
\mathbf{sum}(g)(\vec{x}) &\simeq \begin{cases} \sum_{y\in\mathbb{N}, g(y,\vec{x})\downarrow} g(y,\vec{x}) & \text{if there are only finitely many } y \\ & \qquad \text{with } g(y,\vec{x})\!\downarrow \text{ and } g(y,\vec{x}) > 0, \\ \uparrow & \text{if there are infinitely many } y \\ & \qquad \text{with } g(y,\vec{x})\!\downarrow,\ g(y,\vec{x}) > 0. \end{cases}
\end{aligned}$$

As usual, the empty sum takes the value 0, *i.e.*, $\mathbf{sum}(g)(\vec{x}) = 0$ if all $g(y,\vec{x})$ are undefined. Notice that, possibly, $\mathbf{sum}(g)(\vec{x})\!\downarrow$ even if $g(y,\vec{x})\!\downarrow$ for infinitely many y, but $g(y,\vec{x}) = 0$ in almost all these cases.

Proposition 5.1. *For all* $\boldsymbol{\omega} \in \{\mathbf{max}, \sharp, \mathbf{sum}\}$ *and* $k \geq 1$, $\mathrm{FLev}_{\boldsymbol{\omega}}(k) = \mathrm{FLev}_{\mathbf{lim}}(k) = \mathrm{F}\Sigma_{k+1}$.

To prove this, we show that, applied to functions from $\mathrm{F}\Sigma_k$, any of the operators under considerations, **lim** included, can be expressed by any other of them in the following sense.

Lemma 5.1. *For all* $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \in \{\mathbf{max}, \sharp, \mathbf{sum}, \mathbf{lim}\}$ *and* $\mathrm{FC} \in \{\mathrm{FPaPrim}\} \cup \{\mathrm{F}\Sigma_k : k \geq 1\}$,

$$\boldsymbol{\omega}_1(\mathrm{FC}) \subseteq \mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FC} \cup \boldsymbol{\omega}_2(\mathrm{FC})\,).$$

From this, by means of Proposition 4.1, the equality $\mathrm{FLev}_{\boldsymbol{\omega}}(k) = \mathrm{FLev}_{\mathbf{lim}}(k)$ follows inductively. Indeed, it holds for $k = 0$, and, supposed that it holds for some k, for $\boldsymbol{\omega} \in \{\mathbf{max}, \sharp, \mathbf{sum}\}$ Proposition 4.1 and Lemma 5.1 yield

$$\begin{aligned} \mathrm{FLev}_{\boldsymbol{\omega}}(k+1) &= \mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FLev}_{\boldsymbol{\omega}}(k) \cup \boldsymbol{\omega}(\mathrm{FLev}_{\boldsymbol{\omega}}(k))\,) \\ &= \mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FLev}_{\mathbf{lim}}(k) \cup \mathbf{lim}(\mathrm{FLev}_{\mathbf{lim}}(k))\,) \\ &= \mathrm{FLev}_{\mathbf{lim}}(k+1) = \mathrm{F}\Sigma_{k+2}, \end{aligned}$$

which means the stated equality for $k+1$.

To prove Lemma 5.1, it is enough to show a suitable chain of expressibilities of the operators in the sense of the stated inclusion.

(i) Expressing **lim** by **max**: Let $f = \mathbf{lim}(g)$, for some total function $g \in \mathrm{FC}$ with $\mathrm{FC} \in \{\mathrm{FPaPrim}\} \cup \{\mathrm{F}\Sigma_k :\ k \geq 1\}$. The function g' defined by

$$\begin{aligned} g'(0,\vec{x}) &= 0 \quad \text{and} \\ g'(y+1,\vec{x}) &\simeq \begin{cases} y & \text{if } g(y,\vec{x}) \neq g(y+1,\vec{x}), \\ \uparrow & \text{otherwise,} \end{cases} \end{aligned}$$

belongs to FC too, and it holds $g(\,\mathbf{max}(g')(\vec{x})+1,\vec{x}\,) \simeq \mathbf{lim}(g)(\vec{x})$. Hence $\mathbf{lim}(g)$ belongs to $\mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FC} \cup \mathbf{max}(\mathrm{FC})\,)$.

(ii) Expressing **max** by $\sharp$: For $f = \mathbf{max}(g)$ and $g \in \mathrm{FPaPrim}$, we put

$$g'(y,\vec{x}) \simeq \max\{g(y',\vec{x}) :\ y' \leq y \text{ and } g(y',\vec{x})\!\downarrow\},\quad \text{where } \max\emptyset \simeq\uparrow.$$

Then $g' \in \mathrm{FPaPrim}$, too, and if $g'(y,\vec{x})\!\downarrow$ for some $y \in \mathbb{N}$, then $g'(y',\vec{x})\!\downarrow$ and $g'(y',\vec{x}) \geq g'(y,\vec{x})$ for all $y' \geq y$. By

$$g''(\langle 0,z\rangle,\vec{x}) \quad\simeq\quad \begin{cases} 0 & \text{if } g'(0,\vec{x})\!\downarrow \text{ and } z < g'(0,\vec{x}), \\ \uparrow & \text{otherwise,} \end{cases}$$

$$g''(\langle y+1,z\rangle,\vec{x}) \simeq \begin{cases} 0 & \text{if } g'(y,\vec{x})\!\uparrow \text{ and } z < g'(y+1,\vec{x}) \\ & \text{or } g'(y,\vec{x})\!\downarrow \text{ and } g'(y,\vec{x})+z < g'(y+1,\vec{x}), \\ \uparrow & \text{otherwise,} \end{cases}$$

we get a function $g'' \in \mathrm{FPaPrim}$ such that $\sharp(g'') = \mathbf{max}(g) = f$.

Now let $f = \mathbf{max}(g)$ for some $g \in \mathrm{F}\Sigma_k, k \geq 1$. There is an index n_g with $g = \Phi_{n_g}^{\emptyset^{(k-1)}}$. By

$$g'(y,\vec{x}) \simeq \max\{\Phi_{n_g|y}^{\emptyset^{(k-1)}}(y') : y' \leq y \text{ and } \Phi_{n_g|y}^{\emptyset^{(k-1)}}(y')\!\downarrow\},$$

where $\max\emptyset \simeq\uparrow$, a function $g' \in \mathrm{F}\Sigma_k$ is defined, which satisfies $\mathbf{max}(g) = \mathbf{max}(g')$. Moreover, $\mathrm{dom}(g')$ is even decidable in $\emptyset^{(k-1)}$. If $g'(y,\vec{x})\!\downarrow$ for some $y \in \mathbb{N}$, then $g'(y',\vec{x})\!\downarrow$ and $g'(y',\vec{x}) \geq g'(y,\vec{x})$ for all $y' \geq y$. Defining the function g'' as above for the case $g \in \mathrm{FPaPrim}$, we get $g'' \in \mathrm{F}\Sigma_k$ such that $\sharp(g'') = \mathbf{max}(g) = f$.

(iii) Expressing $\sharp$ by **sum**: This holds quite general. For $f = \sharp(g)$, we put

$$g'(y,\vec{x}) \simeq \begin{cases} 1 & \text{if } g(y,\vec{x})\downarrow, \\ \uparrow & \text{otherwise.} \end{cases}$$

Then $g' \in \mathrm{Clos}_{\{\circ\}}(\,\mathrm{FPaPrim} \cup \{g\}\,)$, and $\mathbf{sum}(g') = \sharp(g) = f$.

(iv) Expressing **sum** by **lim**: Let $f = \mathbf{sum}(g)$. If $g \in \mathrm{FPaPrim}$, we put

$$g'(y,\vec{x}) = \sum\nolimits_{y' \le y, g(y',\vec{x})\downarrow} g(y',\vec{x}).$$

For higher levels, we employ the functions $\Phi^A_{n\|s}$ defined similarly to $\Phi^A_{n|s}$ by

$$\Phi^A_{n\|s}(\vec{x}) \simeq \begin{cases} z & \text{if } \mathcal{M}^A_n(\vec{x}) \text{ halts after exactly } s \text{ steps and} \\ & \text{yields the output } z, \\ \uparrow & \text{otherwise.} \end{cases}$$

If $g \in \mathrm{F}\Sigma_k$, $k \ge 1$ and $g(y,\vec{x}) \simeq \Phi^{\emptyset^{(k-1)}}_{n_g}(\langle y,\vec{x}\rangle)$, let

$$g'(\langle y,z\rangle,\vec{x}) = \sum\nolimits_{\Phi^{\emptyset^{(k-1)}}_{n_g\|z}(y,\vec{x})\downarrow} \Phi^{\emptyset^{(k-1)}}_{n_g\|z}(y,\vec{x}).$$

In all the cases we are considering, we have $g' \in \mathrm{FPaPrim}$ and $g' \in \mathrm{F}\Sigma_k$, respectively, and $\mathbf{lim}(g') = \mathbf{sum}(g) = f$. □

Since the natural numbers are well-ordered, the *minimum operator* **min** cannot simply be seen as an analog to **max**. For any $(m+1)$-ary function g, let

$$\mathbf{min}(g)(\vec{x}) \simeq \min\{g(y,\vec{x}) : g(y,\vec{x})\downarrow\}, \quad \text{where } \min(\emptyset) \simeq \uparrow.$$

Proposition 5.2. *For all* $k \ge 1$, $\mathrm{FLev}_{\mathbf{min}}(k) = \mathrm{FLev}(k+1)$.

Remember that $\mathrm{FLev}(k+1) = \mathrm{FLev}_{\boldsymbol{\omega}}(k+1)$ for all $\boldsymbol{\omega} \in \{\overline{\boldsymbol{\mu}}, \boldsymbol{\phi}, \boldsymbol{\varrho}\}$. The proof of the proposition is by induction on k.

We start with showing that $\mathbf{min}(\mathrm{FPaPrim}) = \overline{\boldsymbol{\mu}}(\mathrm{FCom})$. By $\mathrm{FLev}(1) = \mathrm{FCom} \subseteq \overline{\boldsymbol{\mu}}(\mathrm{FCom})$, this implies

$$\mathrm{FLev}_{\mathbf{min}}(1) = \mathrm{Clos}_{\{\circ\}}(\,\mathrm{FPaPrim} \cup \overline{\boldsymbol{\mu}}(\mathrm{FCom})\,) = \mathrm{FLev}(2).$$

For $f = \mathbf{min}(g)$, $g \in \mathrm{FPaPrim}$, put

$$g'(y,\vec{x}) \simeq \begin{cases} 0 & \text{if there is a } z \in \mathbb{N} \text{ with } g(z,\vec{x}) = y, \\ \uparrow & \text{otherwise.} \end{cases}$$

Then $g' \in \text{FCom}$ and $\overline{\boldsymbol{\mu}}(g') = \mathbf{min}(g) = f$.
For $f = \overline{\boldsymbol{\mu}}(g)$, $g \in \text{FCom}$, suppose that $g = \Phi^{\emptyset}_{n_g}$ and put

$$g'(\langle y, z\rangle, \vec{x}) \simeq \begin{cases} y & \text{if } \Phi^{\emptyset}_{n_g|z}(y, \vec{x}) = 0, \\ \uparrow & \text{otherwise.} \end{cases}$$

Then $g' \in \text{FPaPrim}$ and $\mathbf{min}(g') = \overline{\boldsymbol{\mu}}(g) = f$. So the initial step of induction has been done.

To show that (a single application of) $\overline{\boldsymbol{\mu}}$ can be expressed by $\mathbf{min}$, let $f = \overline{\boldsymbol{\mu}}(g)$, *i.e.*, $f(\vec{x}) \simeq \mathbf{min}\{y : g(y, \vec{x}) = 0\}$. The function g' defined by

$$g'(y, \vec{x}) \simeq \begin{cases} y & \text{if } g(y, \vec{x}) = 0, \\ \uparrow & \text{otherwise,} \end{cases}$$

belongs to the same $\overline{\boldsymbol{\mu}}$-level as g, and it holds $\mathbf{min}(g') = \overline{\boldsymbol{\mu}}(g) = f$. Thus, $\text{FLev}_{\mathbf{min}}(k) = \text{FLev}(k+1)$ implies that

$$\begin{aligned} \text{FLev}(k+2) &= \text{Clos}_{\{\circ\}}\,(\,\text{FLev}(k+1) \cup \overline{\boldsymbol{\mu}}(\text{FLev}(k+1))\,) \\ &\subseteq \text{Clos}_{\{\circ\}}\,(\,\text{FLev}_{\mathbf{min}}(k) \cup \mathbf{min}(\text{FLev}_{\mathbf{min}}(k))\,) \\ &= \text{FLev}_{\mathbf{min}}(k+1). \end{aligned}$$

To prove the converse inclusion, let $f = \mathbf{min}(g)$ for some $g \in \text{FLev}_{\mathbf{min}}(k) = \text{FLev}(k+1)$. If k is even, *i.e.*, $k = 2k'$ for some $k' \in \mathbb{N}$ and $\text{FLev}(k+1) = \text{F}\Sigma_{1+k'}$ by Proposition 3.1, we put

$$g'(z, \vec{x}) \simeq \begin{cases} 0 & \text{if there is a } y \in \mathbb{N} \text{ with } g(y, \vec{x}) = z, \\ \uparrow & \text{otherwise.} \end{cases}$$

This defines a function g' which belongs to $\text{F}\Sigma_{1+k'}$, too, and satisfies $\overline{\boldsymbol{\mu}}(g') = \mathbf{min}(g) = f$. Thus, we have $\mathbf{min}(\text{FLev}(k+1)) \subseteq \overline{\boldsymbol{\mu}}(\text{FLev}(k+1))$, what implies $\text{FLev}_{\mathbf{min}}(k+1) \subseteq \text{FLev}(k+2)$.

If k is odd, *i.e.*, $k = 2k'+1$ with $k' \in \mathbb{N}$, then $\text{FLev}(k+1) = \text{Clos}_{\{\circ\}}\,(\,\text{F}\Sigma_{1+k'} \cup \overline{\boldsymbol{\mu}}(\text{F}\Sigma_{1+k'})\,)$. As noticed in the proof of Proposition 3.1, for all functions g belonging to this class, $\text{dom}(g)$ is decidable in $\emptyset^{(k'+1)}$, and g is computable in $\emptyset^{(k'+1)}$. It follows that the *epigraph* of g,

$$\text{epigraph}(g) = \{(\vec{x}, y) : \text{ there is a } y' \leq y \text{ with } g(\vec{x}) = y'\,\},$$

is decidable in $\emptyset^{(k'+1)}$. Now, if g has some arity ≥ 2, $\mathbf{min}(g)$ can be computed in $\emptyset^{(k'+1)}$ as follows: Given an argument $\vec{x}$, first search for a pair (y, z) such that $((y, \vec{x}), z) \in \text{epigraph}(g)$. If there is such a pair, it will be found finally by a suitable computation in $\emptyset^{(k'+1)}$. Then the value $\mathbf{min}(g)(\vec{x})$ can be determined by deciding whether $((y', \vec{x}), z') \in \text{epigraph}(g)$, for the finitely many $y' \leq y$ and $z' \leq z$. Thus, we have $\mathbf{min}(\mathrm{FLev}(k+1)) \subseteq \mathrm{F}\Sigma_{k'+2}$, and again

$$\begin{aligned}\mathrm{FLev}_{\mathbf{min}}(k+1) &= \mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FLev}(k+1) \cup \mathbf{min}(\mathrm{FLev}(k+1))\,) \\ &\subseteq \mathrm{F}\Sigma_{k'+2} = \mathrm{FLev}(1+2k'+2) = \mathrm{FLev}(k+2).\end{aligned}$$

This completes the inductive proof. □

From the viewpoint of computability theory one might ask why we did not start the nesting levels of operators $\boldsymbol{\omega}$ with $\mathrm{FLev}_{\boldsymbol{\omega}}(1) = \mathrm{FCom}$ and proceed then, as we did, with $\mathrm{FLev}_{\boldsymbol{\omega}}(k+1) = \mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FLev}_{\boldsymbol{\omega}}(k+1) \cup \boldsymbol{\omega}(\mathrm{FLev}_{\boldsymbol{\omega}}(k+1))\,)$. This would not change the levels $\mathrm{FLev}(k)$ of $\overline{\boldsymbol{\mu}}$, $\boldsymbol{\phi}$ and $\boldsymbol{\varrho}$, for $k \geq 1$. The levels $\mathrm{FLev}_{\mathbf{min}}(k)$ would become equal to $\mathrm{FLev}(k)$, however, and for $\boldsymbol{\omega} \in \{\mathbf{lim}, \mathbf{max}, \sharp, \mathbf{sum}\}$ we would have $\mathrm{FLev}_{\boldsymbol{\omega}}(k) = \mathrm{F}\Sigma_k$.

The main reason to start the levels with FPaPrim instead of FCom is that we are highly interested in studying the analogous classification with respect to polynomial-time computability. In our opinion, however, there are more similarities between the classes FP and FPaPrim than between FP and FCom. So it is desirable to know what happens if we start the stratification of levels with FPaPrim.

6. Operators on polytime functions

The theory of computational complexity, in particular that of polynomial-time computability, is mainly devoted to the machine-oriented point of view, where computing devices operate on words over finite alphabets. So the usual complexity classes consist of languages, *i.e.*, sets of words over finite alphabets, and complexity classes of functions usually consist of word functions. The reader is referred to basic textbooks like [2,7,18,23,32]. In the present paper, we continue to deal preferably with sets of numbers and arithmetical functions, respectively. This corresponds to the point of view taken, *e.g.*, in bounded arithmetic, *cf.* [14].

In a certain sense, it is merely a matter of taste that we prefer to deal with explicit number functions and number problems, instead of word functions and word problems. On the one hand, this is here caused by the background from computability theory which has been presented in the preceding sections. On the other hand, most of the operators we are considering refer to a well-ordering of the underlying object domain, which is canonically given for numbers but not so for words. Of course, all the following results and techniques can immediately be transferred to word functions and languages if, between words, we use the order by length and lexicographic comparing based on an ordering of the underlying alphabet.

More precisely, all complexity theoretic notions applied to sets of numbers or number theoretic functions are meant with respect to the (modified) *binary expansion* of numbers, where a *binary word* of *length* $l = |w|$, $w = b_l b_{l-1} \dots b_1 \in \{1,2\}^*$, represents the number

$$\gamma(b_l b_{l-1} \dots b_1) = \sum\nolimits_{i=1}^{l} b_i \cdot 2^{i-1}.$$

The *empty word* Λ represents the number 0, *i.e.*, $\gamma(\Lambda) = 0$. The mapping $\gamma : \{1,2\}^* \longrightarrow \mathbb{N}$ is a bijection between the words from $\{1,2\}^*$ and the natural numbers with respect to which the ordering of words by length and lexicographic comparing corresponds to the natural ordering of numbers. For a number $x \in \mathbb{N}$, $\|x\|$ denotes the *length* of its binary expansion,

$$\|x\| = |\gamma^{-1}(x)|.$$

For an m-tuple $\vec{x} = (x_1, \ldots, x_m)$, let $\|\vec{x}\| = \|\langle x_1, \ldots, x_m\rangle\|$, this is the length of (the binary expansion of) the Cantor number $\langle x_1, \ldots, x_m\rangle$.

A function $f : \mathbb{N}^m \rightarrowtail \mathbb{N}$ is said to be *polynomial-time computable* (briefly: *polytime*) iff the *related word function*, *i.e.*, the mapping $f_{\mathrm{w}}(\{1,2\}^*)^m \rightarrowtail \{1,2\}^*$ defined by

$$f_{\mathrm{w}}(w_1, \ldots, w_m) \simeq \gamma^{-1} \circ f(\,\gamma(w_1), \ldots, \gamma(w_m)\,),$$

is computable in polynomial time by a (deterministic) *Turing machine (TM)* in the usual sense. Let FP denote the class of all polytime functions. For example, the m-tupling function τ^m and its inverses π_i^m, $1 \le i \le m$, are polytime for any $m \ge 1$. Moreover, FP is closed under composition. All polytime functions are *polynomially length-bounded*, *i.e.*, for any $f \in \mathrm{FP}$ there is a polynomial p such that $f(\vec{x}) = y$ implies $\|y\| \le p(\|\vec{x}\|)$.

In the sequel, we shall use the same denotations for some classes of (arithmetical) functions as they occur in literature for related classes of word functions. Also for classes of sets of number tuples, we shall use the same denotations as they are usually applied to the related complexity classes of languages. The denotations of the latter ones will here get the index "w" indicating that they concern (sets of) *words*. So let the class P consist of all sets $A \subseteq \mathbb{N}^m$ that are decidable in polynomial time with respect to the binary expansions of the Cantor numbers, *i.e.*,

$$\mathrm{P} = \{A \subseteq \mathbb{N}^m : \ m \ge 1 \ \text{ and } \ \{\gamma^{-1}\langle x_1, \ldots, x_m\rangle : \ (x_1, \ldots, x_m) \in A\,\} \in \mathrm{P}_{\mathrm{w}}\,\},$$

where P_{w} denotes the usual class of languages decidable by polynomially time-bounded (deterministic) TMs. The classes NP and NP_{w} are analogously related to each other. Of course, $\mathrm{P} = \mathrm{NP}$ iff $\mathrm{P}_{\mathrm{w}} = \mathrm{NP}_{\mathrm{w}}$, and this is the classical P *vs.* NP problem. Also, $A \in \mathrm{P}$ iff $\chi_A \in \mathrm{FP}$.

Polytime functions are partial in general, but, analogously to partial primitive recursive functions, they can be regarded as restrictions of total polytime functions to polynomially decidable domains: For an m-ary function f, it holds $f \in \mathrm{FP}$ iff there are a total function $\widetilde{f} \in \mathrm{FP}$ and a set $A \in \mathrm{P}$, $A \subseteq \mathbb{N}^m$, such that

$$f(\vec{x}) \simeq \begin{cases} \widetilde{f}(\vec{x}) & \text{if } \vec{x} \in A, \\ \uparrow & \text{otherwise.} \end{cases}$$

It follows, *e.g.*, that $\boldsymbol{\phi}(g) \in \mathrm{FCom}$ if $g \in \mathrm{FP}$. On the other hand, the unrestricted application of the first value operator to polytime functions leads already to all computable functions.

Lemma 6.1. *For any* $f \in \mathrm{FCom}$ *there is a function* $g \in \mathrm{FP}$ *such that* $f = \boldsymbol{\phi}(g)$. *Thus, we have* $\boldsymbol{\phi}(\mathrm{FP}) = \mathrm{FCom}$ *and* $\mathrm{Clos}_{\{\circ,\boldsymbol{\phi}\}}(\mathrm{FP}) = \mathrm{FAH}$.

For other operators as powerful as $\boldsymbol{\phi}$, one can prove related results. In fact, instead of FP, more restricted function classes are still sufficient to generate the whole arithmetical hierarchy.

The second part of the lemma follows from $\mathrm{Clos}_{\{\circ,\boldsymbol{\phi}\}}(\mathrm{FCom}) = \mathrm{FAH}$, *cf.* Proposition 3.1. To sketch the proof of the first part, suppose $f = \Phi^{\emptyset}_{n_f}$. We put

$$g(y,\vec{x}) \simeq \begin{cases} \Phi^{\emptyset}_{n_f}(\vec{x}) & \text{if } \Phi^{\emptyset}_{n_f|\|y\|}(\vec{x})\downarrow, \\ \uparrow & \text{otherwise.} \end{cases}$$

It is easily seen that $g \in \mathrm{FP}$ and $f = \boldsymbol{\phi}(g)$. □

In the context of polynomial-time computability, we want to exclude such trivial constructions leading to functions $\boldsymbol{\omega}(g) \notin \mathrm{FP}$ for certain $g \in \mathrm{FP}$. This is done by restricting the search domain for y in a polynomially length-bounded way depending on the remaining argument $\vec{x}$, in building $\boldsymbol{\omega}(g)(\vec{x})$ from the set of all values $g(y,\vec{x})$. A simple but sufficiently general way consists in restricting the operator $\boldsymbol{\omega}$ to functions which are undefined for arguments $(y,\vec{x})$ if y lies outside a polynomial length-bound depending on $\vec{x}$.

More precisely, an $(m+1)$-ary function g is said to be 1-*polynomial* iff there is a polynomial p such that

$$g(y,\vec{x})\uparrow \quad \text{for all } y \in \mathbb{N} \text{ and } \vec{x} \in \mathbb{N}^m \text{ satisfying } \|y\| > p(\|\vec{x}\|).$$

In other words, for all numbers y with $g(y,\vec{x})\downarrow$ we have $0 \leq y < 2^{p(\|\vec{x}\|)} - 1$. It follows that, in computing $\boldsymbol{\phi}(g)(\vec{x})$, the search for a minimal y with $g(y,\vec{x})\downarrow$ can be restricted to the set $\{y : 0 \leq y < 2^{p(\|\vec{x}\|)} - 1\}$.

Let $\boldsymbol{\omega}'$ denote the restriction of an operator $\boldsymbol{\omega} \in \{\boldsymbol{\mu}, \overline{\boldsymbol{\mu}}, \boldsymbol{\phi}, \mathbf{max}, \sharp, \mathbf{sum}, \mathbf{min}\}$ to 1-polynomial functions of arities ≥2. More precisely, $\boldsymbol{\omega}'(g)$ is defined iff g is 1-polynomial and $\boldsymbol{\omega}(g)$ is defined, and in this case we put $\boldsymbol{\omega}'(g) = \boldsymbol{\omega}(g)$. For example, if g is 1-polynomial, then both $\sharp'(g)$ and $\mathbf{sum}'(g)$ are total functions. Since the analogous restriction $\mathbf{lim}'$ yields only the nowhere defined function, it is not of interest.

For an adequate restriction of the operator $\boldsymbol{\varrho}$, we employ a notion well known in complexity theory. A function $g : \mathbb{N} \rightarrowtail \mathbb{N}$ is said to be (polynomially) *honest* iff there is a polynomial p such that

$$\boldsymbol{\varrho}(g)(y) = x \quad \text{implies} \quad \|x\| \leq p(\|y\|).$$

This condition is obviously necessary for $\boldsymbol{\varrho}(g) \in \mathrm{FP}$.

Let $\boldsymbol{\varrho}'$ denote the restriction of the operator $\boldsymbol{\varrho}$ to honest (unary) functions.

It is easily seen that all functions from $\mathrm{Clos}_{\{\circ,\boldsymbol{\omega}'\}}(\mathrm{FP})$ are polynomially length-bounded, for all $\boldsymbol{\omega}' \in \{\boldsymbol{\mu}', \overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}', \mathbf{max}', \sharp', \mathbf{sum}', \mathbf{min}'\}$.

The *polytime* $\boldsymbol{\omega}'$-*levels* $\mathrm{FLev}'_{\boldsymbol{\omega}'}(k)$, for numbers $k \in \mathbb{N}$, are defined in a straightforward way similarly to the $\boldsymbol{\omega}$-levels. In contrast to them, however, we now start with the basic class FP:

$$\begin{aligned}\mathrm{FLev}'_{\boldsymbol{\omega}'}(0) &= \mathrm{FP} \quad \text{and}\\ \mathrm{FLev}'_{\boldsymbol{\omega}'}(k+1) &= \mathrm{Clos}_{\{\circ\}}\,(\ \mathrm{FLev}'_{\boldsymbol{\omega}'}(k) \cup \boldsymbol{\omega}'(\mathrm{FLev}'_{\boldsymbol{\omega}'}(k))\ ) \quad \text{for } k \in \mathbb{N}.\end{aligned}$$

First it turns out that, in analogy to Proposition 2.1, the operators $\overline{\boldsymbol{\mu}}'$, $\boldsymbol{\phi}'$ and $\boldsymbol{\varrho}'$ yield the same polytime levels.

Proposition 6.1. *For any* $\boldsymbol{\omega}'_1, \boldsymbol{\omega}'_2 \in \{\overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}'\}$ *and* $k \in \mathbb{N}$, *we have* $\mathrm{FLev}'_{\boldsymbol{\omega}'_1}(k) = \mathrm{FLev}'_{\boldsymbol{\omega}'_2}(k)$.

The proof is quite analogous to that of Proposition 2.1, namely based on the following lemma stating the mutual expressibilities of the operators under consideration.

Lemma 6.2. *For* $\boldsymbol{\omega}'_1, \boldsymbol{\omega}'_2 \in \{\overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}'\}$ *and every function class* FC *which is closed under composition and satisfies* $\mathrm{FP} \subseteq \mathrm{FC} \subseteq \mathrm{FAll}$, *we have*

$$\boldsymbol{\omega}'_1(\mathrm{FC}) \subseteq \mathrm{Clos}_{\{\circ\}}\,(\ \mathrm{FC} \cup \boldsymbol{\omega}'_2(\mathrm{FC})\ ).$$

This can be shown by almost the same constructions as in the proof of Lemma 2.1, parts (i), (ii) and (iii). One now has to notice that if the functions g are 1-polynomial and honest, respectively, the functions g' defined in the related ways are honest or 1-polynomial, too. Only in part (iii), the definition of function g' has to be modified to

$$g'(y,z) \simeq \begin{cases} 0 & \text{if } g(y) = z \text{ and } \|y\| \leq p(\|z\|),\\ \uparrow & \text{otherwise,}\end{cases}$$

with a polynomial p witnessing that the function g is honest. In all the three parts, it holds $g' \in \mathrm{Clos}_{\{\circ\}}\,(\ \mathrm{FP} \cup \{g\}\ ) \subseteq \mathrm{FC}$ and $\boldsymbol{\omega}'_1(g) \in \mathrm{Clos}_{\{\circ\}}\,(\mathrm{FC} \cup \{\boldsymbol{\omega}'_2(g')\}\,)$, for the related operators $\boldsymbol{\omega}'_1, \boldsymbol{\omega}'_2 \in \{\overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}'\}$. So the proof of Lemma 6.2 and Proposition 6.1 is complete. ☐

In the sequel, we shall simply write $\mathrm{FLev}'(k)$ instead of $\mathrm{FLev}'_{\boldsymbol{\omega}'}(k)$ with $\boldsymbol{\omega}' \in \{\overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}'\}$.

In contrast to the results of Section 5 concerning general computability, the polytime variants of the operators **max** and **min** are equivalent to each other and even to $\overline{\boldsymbol{\mu}}', \boldsymbol{\phi}'$ and $\boldsymbol{\varrho}'$.

Proposition 6.2. *For any* $\boldsymbol{\omega}'_1, \boldsymbol{\omega}'_2 \in \{\mathbf{min}', \mathbf{max}', \overline{\boldsymbol{\mu}}'\}$ *and every function class* FC *which is closed under composition and includes* FP, *we have* $\boldsymbol{\omega}'_1(\mathrm{FC}) \subseteq \mathrm{Clos}_{\{\circ\}}\,(\ \mathrm{FP} \cup \boldsymbol{\omega}'_2(\mathrm{FC})\ )$. *Thus,* $\mathrm{FLev}'_{\mathbf{min}'}(k) = \mathrm{FLev}'_{\mathbf{max}'}(k) = \mathrm{FLev}'(k)$ *for all* $k \in \mathbb{N}$.

The second statement follows from the first one analogously to the proof of Proposition 6.1.

Now suppose that g is an $(m+1)$-ary function which is 1-polynomial with respect to a polynomial p, *i.e.*, $g(y,\vec{x})\uparrow$ whenever $\|y\| > p(\|\vec{x}\|)$. First we put

$$g'(y,\vec{x}) \simeq 2^{p(\|\vec{x}\|)} - g(y,\vec{x}).$$

Then g' is 1-polynomial too, and $g' \in \mathrm{Clos}_{\{\circ\}}(\ \mathrm{FP} \cup \{g\}\ )$. Moreover, for all $\vec{x} \in \mathbb{N}^m$,

$$\mathbf{max}'(g)(\vec{x}) \simeq 2^{p(\|\vec{x}\|)} - \mathbf{min}'(g')(\vec{x}) \quad \text{and} \quad \mathbf{min}'(g)(\vec{x}) \simeq 2^{p(\|\vec{x}\|)} - \mathbf{max}'(g')(\vec{x}).$$

This shows the mutual expressibility of $\mathbf{max}'$ and $\mathbf{min}'$ by means of polytime functions.

Putting

$$g'(y,\vec{x}) \simeq \begin{cases} y & \text{if } g(y,\vec{x}) = 0, \\ \uparrow & \text{otherwise,} \end{cases}$$

we get a 1-polynomial function $g' \in \mathrm{Clos}_{\{\circ\}}(\ \mathrm{FP} \cup \{g\}\ )$ with $\mathbf{min}'(g') = \overline{\boldsymbol{\mu}}'(g)$.

On the other hand, if $g(y,\vec{x})\uparrow$ whenever $\|y\| > p(\|\vec{x}\|)$, by

$$g'(y \cdot 2^{p(\|\vec{x}\|)} + z, \vec{x}) \simeq \begin{cases} 0 & \text{if } g(y,\vec{x}) = z, \\ \uparrow & \text{otherwise,} \end{cases}$$

we again get a 1-polynomial function $g' \in \mathrm{Clos}_{\{\circ\}}(\ \mathrm{FP} \cup \{g\}\ )$. Now it holds

$$\mathbf{min}'(g)(\vec{x}) \simeq \overline{\boldsymbol{\mu}}'(g') \bmod 2^{p(\vec{x})},$$

where mod denotes the binary modulus function, *i.e.*, $a \bmod b$ is the remainder of a divided by b, for $a,b \in \mathbb{N}$ and $b \geq 1$. Notice that $\bmod \in \mathrm{FP}$. □

The discussion of the power of the operators $\sharp', \mathbf{sum}'$ and $\boldsymbol{\mu}'$ requires knowledge of some relationships to the polynomial hierarchy. Hence it is postponed to Sections 7 and 9.

7. First relationships to the polynomial hierarchy and P *vs.* NP

Analogously to the arithmetical hierarchy, the (number theoretic version of the) *polynomial hierarchy* is considered both as the hierarchy of classes Σ_k^p, Π_k^p and Δ_k^p, for $k \in \mathbb{N}$, but also as the union over them: $\mathrm{PH} = \bigcup_{k=0}^{\infty} \Sigma_k^p$. To characterize the classes Σ_k^p, one can use *bounded quantifications*, where

$$\begin{array}{lll} \forall^m z \ldots & \text{stands for} & \forall z(\ \|z\| \leq m \Rightarrow \ \ldots\ ) \ \text{ and} \\ \exists^m z \ldots & \text{stands for} & \exists z(\ \|z\| \leq m \ \wedge \ \ldots\ ). \end{array}$$

Σ_k^p consists of all relations $A \subseteq \mathbb{N}^m$, $m \geq 1$, which are representable in the form

$$A = \{\vec{x} \in \mathbb{N}^m : \exists^{p_1(\|\vec{x}\|)} y_1 \, \forall^{p_2(\|\vec{x}\|)} y_2 \ldots Q^{p_k(\|\vec{x}\|)} y_k \ (y_1, \ldots, y_k, \vec{x}) \in B\}$$

with $B \in \mathrm{P}$, polynomials $p_1, \ldots, p_k$, and $Q \in \{\exists, \forall\}$, so that the prefix of quantifiers becomes alternating. One could (equivalently) require that all polynomials occurring above coincide: $p_1 = \cdots = p_k = p$ for some polynomial p. As usual, let $\Pi^p_k = \mathrm{co}\Sigma^p_k$, *i.e.*, it consists of the complements of the members of Σ^p_k; the classes Δ^p_k, however, are not defined as the intersections of Σ^p_k with Π^p_k, see below. In particular, $\Sigma^p_0 = \Pi^p_0 = \Delta^p_0 = \Delta^p_1 = \mathrm{P}$, $\Sigma^p_1 = \mathrm{NP}$ and $\Pi^p_1 = \mathrm{coNP}$. It is known that a relation $A \subseteq \mathbb{N}^m$ belongs to PH iff it is representable by a formula of *bounded arithmetic*, see [14] for related details.

Another characterization of (the classes of) PH can be given by means of *deterministic and nondeterministic polynomial-time oracle Turing machines* (*POTM* and *NPOTM*, respectively). For a (complexity) class of sets, C, let P^C and NP^C denote the class of all sets $A \subseteq \mathbb{N}^m, m \geq 1$, which can be accepted by POTMs and NPOTMs, respectively, using oracle sets belonging to C. Then

$$\Sigma^p_0 = \Pi^p_0 = \Delta^p_0 = \mathrm{P};\ \ \Sigma^p_{k+1} = \mathrm{NP}^{\Sigma^p_k},\ \text{and}\ \Pi^p_k = \mathrm{co}\Sigma^p_k \quad \text{for all } k \in \mathbb{N}.$$

The classes Δ^p_{k+1} are usually defined in a special way:

$$\Delta^p_{k+1} = \mathrm{P}^{\Sigma^p_k}.$$

For motivations, further details and basic results, the reader is referred to textbooks of complexity theory.

The *function classes* FP^C and FNP^C, for a complexity class C of sets, correspondingly consist of all functions computable by POTMs and NPOTMs, respectively, which use oracle sets from C. The meaning should be clear for the deterministic case. *Computation of a function by a nondeterministic machine* means that, for any input, all its terminating computations yield the same result, namely the function value at this input, and that there is at least one such terminating computation iff the function is defined at this input. Equivalently one can say that $f \in \mathrm{FNP}^C$ iff there is a 1-polynomial function $g \in \mathrm{FP}^C$ such that

i) $g(y_1, \vec{x}) = g(y_2, \vec{x})$ whenever $g(y_1, \vec{x})\downarrow$ and $g(y_2, \vec{x})\downarrow$, and

ii) $f = \phi'(g)$.

So we have $\mathrm{FP}^C \subseteq \mathrm{FNP}^C$, $A \in \mathrm{P}^C$ iff $\chi_A \in \mathrm{FP}^C$, and $A \in \mathrm{NP}^C$ iff $\chi^0_A \in \mathrm{FNP}^C$, where χ_A denotes the (total) *characteristic function* of a set A, in contrast to its (partial) semicharacteristic function χ^0_A, *cf.* Section 3.

Generalizing the definition of the classes $\mathrm{F}\Sigma_k$ in the arithmetical hierarchy, any class C consisting of sets determines a function class $\mathrm{F}[C] = \{f \in \mathrm{FAll} : \mathrm{graph}(f) \in C\}$. With respect to polytime computability, it is natural to restrict these classes to polynomially length-bounded functions. Thus, we put

$$\mathrm{F}_p[C] = \{f \in \mathrm{FAll} :\ \mathrm{graph}(f) \in C \text{ and } f \text{ is polynomially length-bounded}\}.$$

Lemma 7.1. *For every class C of sets, we have* $\mathrm{F}_p[\mathrm{NP}^C] = \mathrm{FNP}^C$.

If graph$(f) \in \mathrm{NP}^C$, *i.e.*, $\chi^0_{\mathrm{graph}(f)} \in \mathrm{FNP}^C$, it holds $\chi^0_{\mathrm{graph}(f)} = \boldsymbol{\phi}'(g)$ for some $g \in \mathrm{FP}^C$ satisfying the above uniqueness property i). Putting

$$h(\langle y, z\rangle, \vec{x}) \simeq \begin{cases} z & \text{if } g(y, \vec{x}, z)\downarrow, \\ \uparrow & \text{otherwise,} \end{cases}$$

we get a function $h \in \mathrm{FP}^C$ such that $h(\langle y_1, z_1\rangle, \vec{x}) = h(\langle y_2, z_2\rangle, \vec{x})$ whenever $h(\langle y_1, z_1\rangle, \vec{x})\downarrow$ and $h(\langle y_2, z_2\rangle, \vec{x})\downarrow$, and $f = \boldsymbol{\phi}(h)$. If f is polynomially length-bounded, then h is 1-polynomial, hence $f = \boldsymbol{\phi}'(h)$ and $f \in \mathrm{FNP}^C$.

Conversely, if $f = \boldsymbol{\phi}'(g)$ with $g \in \mathrm{FP}^C$ satisfying condition i), we have

$$\chi^0_{\mathrm{graph}(f)}(\vec{x}, z) \simeq \begin{cases} 0 & \text{if } \boldsymbol{\phi}'(g)(\vec{x}) = z, \\ \uparrow & \text{otherwise.} \end{cases}$$

Thus, graph$(f) \in \mathrm{NP}^C$. Moreover, f is polynomially length-bounded. □

It should be noticed that the analogue for the deterministic case does not hold, probably. For example, we have $\mathrm{F}_p[\mathrm{P}] = \mathrm{FP}$ iff $\mathrm{P} = \mathrm{UP}$. This is a result by Grollman and Selman, *cf.* Satz 9.7 in [32].

Of course, it holds $\mathrm{FP}^{\mathrm{P}} = \mathrm{FP}$. The class $\mathrm{FNP}^{\mathrm{P}}$ contains exactly those functions which are computable by NPOTMs with the empty oracle set or, equivalently, without using the oracle. This class will also be denoted by FNP, and its elements are called *nondeterministically computable in polynomial time*. FNP is just the number theoretic analogue of Selman's class NPSV, see [25,26]. Lemma 7.1 can also be seen as an analogue of the (relativized version of the) well-known graph Theorem of computability: $f \in \mathrm{FCom}$ iff graph(f) is c.e. From this point of view, the lemma and the remark above indicate that FCom is more related to FNP than to FP, whereas the latter could better be seen as a counterpart of FPaPrim in computability theory, *cf.* our remark at the end of Section 5.

Anyway, it is natural to define the following *function classes of the polynomial hierarchy*:

$$\mathrm{F}\Sigma^p_k = \mathrm{F}_p[\Sigma^p_k] \quad \text{for } k \in \mathbb{N}.$$

So we have $\mathrm{FP} \subseteq \mathrm{F}_p[P] = \mathrm{F}\Sigma^p_0$, $\mathrm{F}\Sigma^p_1 = \mathrm{F}_p[\mathrm{NP}] = \mathrm{FNP}$, and

$$\mathrm{F}\Sigma^p_{k+1} = \mathrm{F}_p[\Sigma^p_{k+1}] = \mathrm{F}_p[\mathrm{NP}^{\Sigma^p_k}] = \mathrm{FNP}^{\Sigma^p_k} \quad \text{for all } k \in \mathbb{N}.$$

Let FPH denote the class of all polynomially length-bounded functions whose graphs belong to PH, *i.e.*,

$$\mathrm{FPH} = \mathrm{F}_p[\mathrm{PH}] = \bigcup_{k \in \mathbb{N}} \mathrm{F}\Sigma^p_k.$$

By means of the characterization of the classes Σ^p_k by bounded quantifications, it is easily seen that the property of belonging to FPH is hereditary both under the composition of functions and under the application of $\overline{\boldsymbol{\mu}}'$. Thus, we have $\mathrm{Clos}_{\{\circ, \overline{\boldsymbol{\mu}}'\}}(\mathrm{FP}) \subseteq \mathrm{FPH}$ and, by Propositions 6.1 and 6.2, $\mathrm{Clos}_{\{\circ, \boldsymbol{\omega}'\}}(\mathrm{FP}) \subseteq \mathrm{FPH}$

for all $\boldsymbol{\omega}' \in \{\overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}', \mathbf{min}', \mathbf{max}'\}$. To give a first proof of the converse inclusion, we show

Lemma 7.2. *If* $A \in \mathrm{PH}$*, then* $\chi_A \in \mathrm{Clos}_{\{\mathsf{o},\overline{\boldsymbol{\mu}}'\}}(\mathrm{FP})$.

This is proved by induction on k, for all $A \in \Sigma_k^p$. If $k = 0$, *i.e.*, $A \in \mathrm{P}$, then $\chi_A \in \mathrm{FP}$.

Suppose the assertion holds for all $A \in \Sigma_k^p$. First it follows for all $A \in \Pi_k^p$ too, since $\chi_A(\vec{x}) = 1 - \chi_{\overline{A}}(\vec{x})$, hence $\chi_A \in \mathrm{Clos}_{\{\mathsf{o}\}}(\mathrm{FP} \cup \{\chi_{\overline{A}}\})$.

Now let $A \in \Sigma_{k+1}^p$, *i.e.*,

$$\vec{x} \in A \quad \text{iff} \quad \exists^{p_0(\|\vec{x}\|)} y_0\, \forall^{p_1(\|\vec{x}\|)} y_1 \ldots Q^{p_k(\|\vec{x}\|)} y_k\, (y_0, y_1, \ldots, y_k, \vec{x}) \in B,$$

with $B \in \mathrm{P}$, polynomials $p_0, \ldots, p_k$ and a related quantifier Q. By the supposition, for the set $A' \in \Pi_k^p$ defined by

$$(y_0, \vec{x}) \in A' \quad \text{iff} \quad \forall^{p_1(\|\vec{x}\|)} y_1 \ldots Q^{p_k(\|\vec{x}\|)} y_k\, (y_0, y_1, \ldots, y_k, \vec{x}) \in B,$$

we have $\chi_{A'} \in \mathrm{Clos}_{\{\mathsf{o},\overline{\boldsymbol{\mu}}'\}}(\mathrm{FP})$. Putting

$$g(y_0, \vec{x}) \simeq \begin{cases} 1 - \chi_{A'}(y_0, \vec{x}) & \text{if } \|y_0\| \leq p_0(\|\vec{x}\|), \\ 0 & \text{if } \|y_0\| = p_0(\|\vec{x}\|) + 1, \\ \uparrow & \text{otherwise,} \end{cases}$$

we get a 1-polynomial function $g \in \mathrm{Clos}_{\{\mathsf{o},\overline{\boldsymbol{\mu}}'\}}(\mathrm{FP})$ with $\chi_A(\vec{x}) = h(\overline{\boldsymbol{\mu}}'(g)(\vec{x}), \vec{x})$ for the function $h \in \mathrm{FP}$ defined by

$$h(y, \vec{x}) = \begin{cases} 1 & \text{if } \|y\| \leq p_0(\|\vec{x}\|), \\ 0 & \text{otherwise.} \end{cases}$$

□

Proposition 7.1. *For every* $\boldsymbol{\omega}' \in \{\overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}', \mathbf{min}', \mathbf{max}'\}$*,* $\mathrm{Clos}_{\{\mathsf{o},\boldsymbol{\omega}'\}}(\mathrm{FP}) = \mathrm{FPH}$.

To show the remaining part of the proof, *i.e.*, $\mathrm{FPH} \subseteq \mathrm{Clos}_{\{\mathsf{o},\overline{\boldsymbol{\mu}}'\}}(\mathrm{FP})$, let $\mathrm{graph}(f) \in \mathrm{PH}$ and $\|f(\vec{x})\| \leq p(\|\vec{x}\|)$ for all $\vec{x} \in \mathrm{dom}(f)$, with a polynomial p. Then it holds $f(\vec{x}) = y$ iff $1 - \chi_{\mathrm{graph}(f)}(\vec{x}, y) = 0$. So we have $f = \overline{\boldsymbol{\mu}}'(g)$ for

$$g(y, \vec{x}) \simeq \begin{cases} 1 - \chi_{\mathrm{graph}(f)}(\vec{x}, y) & \text{if } \|y\| \leq p(\|\vec{x}\|), \\ \uparrow & \text{otherwise.} \end{cases}$$

By Lemma 7.2, it follows that $f \in \mathrm{Clos}_{\{\mathsf{o},\overline{\boldsymbol{\mu}}'\}}(\mathrm{FP})$. □

It is well-known that $\mathrm{P} = \mathrm{NP}$ iff $\mathrm{P} = \mathrm{PH}$ or, equivalently, $\mathrm{P} = C$ for some (or all) class(es) C with $\mathrm{NP} \subseteq C \subseteq \mathrm{PH}$. Other equations equivalent to $\mathrm{P} = \mathrm{NP}$ are $\mathrm{FP} = \mathrm{FNP}$, $\mathrm{FP} = \mathrm{FPH}$, and $\mathrm{FP} = \mathrm{FC}$ for some (or all) FC satisfying $\mathrm{FNP} \subseteq \mathrm{FC} \subseteq \mathrm{FPH}$. This was shown in [25,26] and seems to be folklore. Proposition 7.1 immediately yields the following characterization of the P *vs.* NP problem in terms of a closure property of FP with respect to certain function operators.

Theorem 7.1. *For every* $\boldsymbol{\omega}' \in \{\overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}', \mathbf{min}', \mathbf{max}'\}$ *and any other operator* $\boldsymbol{\omega}'$ *which satisfies* $\mathrm{FNP} \subseteq \mathrm{Clos}_{\{\circ,\boldsymbol{\omega}'\}}(\mathrm{FP}) \subseteq \mathrm{FPH}$*, we have*

$$\mathrm{P} = \mathrm{NP} \quad \textit{iff} \quad \mathrm{FP} \textit{ is closed under } \boldsymbol{\omega}'.$$

Indeed, P = NP implies the closedness of FP under $\boldsymbol{\omega}'$ by the remarks above. Conversely, if FP is closed under $\boldsymbol{\omega}'$, then $\mathrm{FPH} = \mathrm{Clos}_{\{\circ,\boldsymbol{\omega}'\}}(\mathrm{FP}) = \mathrm{FP}$, since FP is closed under the composition of functions. □

Now it is time to consider $\boldsymbol{\mu}'$, the restriction of the classical $\boldsymbol{\mu}$-operator to 1-polynomial functions, in the present context. Using the characterization of the classes Σ_k^p by bounded quantifications, one easily sees that $\mathrm{Clos}_{\{\circ,\boldsymbol{\mu}'\}}(\mathrm{FP}) \subseteq \mathrm{FPH}$. On the other hand, let $f \in \mathrm{FNP}$, *i.e.*, $f = \boldsymbol{\phi}'(g)$ for $g \in \mathrm{FP}$ such that $g(y, \|\vec{x}\|)\uparrow$ whenever $\|y\| > p(\|\vec{x}\|)$, with a suitable polynomial p. For

$$g'(y,\vec{x}) \simeq \begin{cases} 0 & \text{if } g(y,\vec{x})\downarrow, \\ 1 & \text{if } g(y,\vec{x})\uparrow \text{ and } \|y\| \le p(\|\vec{x}\|), \\ \uparrow & \text{if } \|y\| > p(\|\vec{x}\|), \end{cases}$$

we have $g' \in \mathrm{FP}$ and $f(\vec{x}) \simeq g(\boldsymbol{\mu}'(g')(\vec{x}), \vec{x})$, hence $f \in \mathrm{Clos}_{\{\circ,\boldsymbol{\mu}'\}}(\mathrm{FP})$ and even $f \in \mathrm{FLev}_{\boldsymbol{\mu}'}(1)$. This would already suffice to show that Theorem 7.1 applies to $\boldsymbol{\omega}' = \boldsymbol{\mu}'$ too. Moreover, the proofs of Lemma 7.2 and Proposition 7.1 work also with $\boldsymbol{\mu}'$ instead of $\overline{\boldsymbol{\mu}}'$. Thus, $\mathrm{Clos}_{\{\circ,\boldsymbol{\mu}'\}}(\mathrm{FP}) = \mathrm{FPH}$, and it follows

Corollary 7.1. $\mathrm{Clos}_{\{\circ,\boldsymbol{\mu}'\}}(\mathrm{FP}) = \mathrm{FPH}$*, and it holds* P = NP *iff* FP *is closed under* $\boldsymbol{\mu}'$. □

Next we turn to a closer treatment of the nesting levels of (the polytime versions of) some operators within FPH.

8. The polytime levels within the polynomial hierarchy

As for the polynomial hierarchy of sets, it is useful to enrich the polynomial hierarchy of functions by classes $\mathrm{F}\Delta_{k+1}^p$ defined by means of POTMs in analogy to the employment of NPOTMs in defining $\mathrm{F}\Sigma_{k+1}^p = \mathrm{FNP}^{\Sigma_k^p}$, *cf.* Lemma 7.1. So we put

$$\mathrm{F}\Delta_{k+1}^p = \mathrm{FP}^{\Sigma_k^p} \quad \text{for all } k \in \mathbb{N}.$$

In particular, $\mathrm{F}\Delta_1^p = \mathrm{FP}^{\Sigma_0^p} = \mathrm{FP}$, and $\mathrm{F}\Delta_2^p = \mathrm{FP}^{\Sigma_1^p} = \mathrm{FP}^{\mathrm{NP}}$. From $\mathrm{F}\Sigma_1^p = \mathrm{FNP}$ upwards, we have the usual inclusions:

Lemma 8.1. *For all* $k \ge 1$, $\mathrm{F}\Sigma_k^p \subseteq \mathrm{F}\Delta_{k+1}^p \subseteq \mathrm{F}\Sigma_{k+1}^p$.

The second inclusion is trivial. To prove the first one, let $f \in \mathrm{F}\Sigma_k^p = \mathrm{F}_p[\Sigma_k^p]$. This means that $\mathrm{graph}(f) \in \Sigma_k^p$, and f is polynomially length-bounded. Since $k \ge 1$, the set

$$A_f = \{(y_1, y_2, \vec{x}) : \exists y\,(\, y_1 \le y \wedge y \le y_2 \wedge f(\vec{x}) = y\,)\}$$

belongs to Σ^p_k too: the quantification $\exists y$ can be polynomially length-bounded with respect to the tuple $\vec{x}$, and then it can be joined with the first existential quantification in the presentation of graph(f) characterizing $f(\vec{x}) = y$ which is also polynomially length-bounded with respect to the argument $\vec{x}$.

Finally, by means of the oracle set A_f a POTM can, for any given input $\vec{x}$, decide whether $\vec{x} \in \mathrm{dom}(f)$, and if so, it can compute the value $f(\vec{x})$ by binary search, again using the oracle set A_f. □

Now we are going to show that the levels of our operators coincide with the classes $\mathrm{F}\Delta^p_{k+1}$.

Theorem 8.1. *For all* $k \in \mathbb{N}$*,* $\mathrm{FLev}'(k) = \mathrm{F}\Delta^p_{k+1}$*.*

The proof is by induction on k. For $k = 0$, we indeed have

$$\mathrm{FLev}'(0) = \mathrm{FP} = \mathrm{FP}^{\mathrm{P}} = \mathrm{FP}^{\Sigma^p_0} = \mathrm{F}\Delta^p_1.$$

Supposed that the assertion holds for some k, it follows for $k+1$, *i.e.*,

$$\mathrm{FLev}'(k+1) = \mathrm{F}\Delta^p_{k+2}.$$

To show inclusion "$\subseteq$", we remember that $\mathrm{FLev}'(k+1) = \mathrm{Clos}_{\{\circ\}}(\mathrm{FLev}'(k) \cup \boldsymbol{\phi}'(\mathrm{FLev}'(k)))$. Since $\mathrm{FLev}'(k) \subseteq \boldsymbol{\phi}'(\mathrm{FLev}'(k))$ and $\mathrm{F}\Delta^p_{k+2}$ is closed under composition, it is enough to show that $\boldsymbol{\phi}'(\mathrm{FLev}'(k)) = \boldsymbol{\phi}'(\mathrm{F}\Delta^p_{k+1}) \subseteq \mathrm{F}\Delta^p_{k+2}$, *i.e.*,

$$\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma^p_k}) \subseteq \mathrm{FP}^{\Sigma^p_{k+1}}.$$

Given a 1-polynomial function $g \in \mathrm{FP}^{\Sigma^p_k}$, we put

$$A_g = \{(y_1, y_2, \vec{x}) : \exists y(\, y_1 \le y \wedge y \le y_2 \wedge g(y, \vec{x})\!\downarrow)\}.$$

Obviously, $A_g \in \mathrm{NP}^{\Sigma^p_k} = \Sigma^p_{k+1}$. As in the proof of Lemma 8.1, for any tuple $\vec{x}$ by binary search using the oracle set A_g, a POTM can decide whether there exists a y with $g(y, \vec{x})\!\downarrow$, and if so, the minimal such y can be determined. Hence we have $\boldsymbol{\phi}'(g) \in \mathrm{FP}^{\Sigma^p_{k+1}}$.

It should be noticed that one can even show $\boldsymbol{\phi}'(\mathrm{FNP}^{\Sigma^p_k}) \subseteq \mathrm{FP}^{\Sigma^p_{k+1}}$. Indeed, also for a 1-polynomial function $g \in \mathrm{FNP}^{\Sigma^p_k}$ it follows $A_g \in \mathrm{NP}^{\Sigma^p_k}$.

To prove the converse inclusion $\mathrm{F}\Delta^p_{k+2} \subseteq \mathrm{FLev}'(k+1)$ by means of the inductive hypothesis, the following lemma is crucial.

Lemma 8.2. *Every total function* $f \in \mathrm{F}\Delta^p_{k+2}$ *belongs to* $\boldsymbol{\phi}'(\mathrm{F}\Delta^p_{k+1})$*.*

Supposed that the lemma holds and $f \in \mathrm{F}\Delta^p_{k+2}$, we consider its *totalization*,

$$\overline{f}(\vec{x}) = \begin{cases} f(\vec{x}) + 1 & \text{if } f(\vec{x})\downarrow, \\ 0 & \text{if } f(\vec{x})\uparrow. \end{cases}$$

Then $\overline{f} \in \mathrm{F}\Delta^p_{k+2}$ too, and Lemma 8.2 yields $\overline{f} \in \phi'(\mathrm{F}\Delta^p_{k+1})$. On the other hand, $f = h_0 \circ \overline{f}$ with the unary function $h_0 \in \mathrm{FP}$ defined by

$$h_0(z) \simeq \begin{cases} z-1 & \text{if } z > 0, \\ \uparrow & \text{if } z = 0. \end{cases}$$

Thus, the hypothesis of induction would imply

$$f \in \mathrm{Clos}_{\{\circ\}}\,(\,\phi'(\mathrm{F}\Delta^p_{k+1})\,) = \mathrm{Clos}_{\{\circ\}}\,(\,\phi'(\mathrm{FLev}'(k))\,) \subseteq \mathrm{FLev}'(k+1),$$

and the proof of Theorem 8.1 would be complete.

Lemma 8.2 can be proved by adapting a technique established by Krentel [15] in his proof that $\mathrm{MAX\,P} \supseteq \Delta^p_2$, see also [32].

Let be given a total m-ary function $f \in \mathrm{FP}^{\Sigma^p_{k+1}}$ which is computed by a POTM $\mathcal{M}$ using an oracle set $A \in \Sigma^p_{k+1} = \mathrm{NP}^{\Sigma^p_k}$. Without loss of generality, $A \subseteq \mathbb{N}$. Let p denote a polynomial time bound of $\mathcal{M}$, *i.e.*, if $f(\vec{x})\downarrow$, then $\mathcal{M}$ halts after at most $p(\|\vec{x}\|)$ steps on input $\vec{x}$. Since $\chi^0_A \in \mathrm{FNP}^{\Sigma^p_k}$, there is a 1-polynomial binary function $h \in \mathrm{FP}^{\Sigma^p_k}$ such that

$$A = \{z \mid \exists^{p_A(\|z\|)} y\, h(y,z) = 0\,\}, \qquad \text{with a suitable polynomial } p_A.$$

It can be supposed that $\mathrm{ran}(h) \subseteq \{0,1\}$ and $h(y,z)\downarrow$ whenever $\|y\| \leq p_A(\|z\|)$.

Now we describe the work of a POTM $\mathcal{M}'$ which computes a 1-polynomial (partial) function $g : \mathbb{N}^{m+1} \rightarrowtail \mathbb{N}$. Given an input $(y,\vec{x}) \in \mathbb{N}^{m+1}$ with $\|y\| \leq (\,p_A \circ p(\|\vec{x}\|) + 1\,) \cdot p(\|\vec{x}\|)$, its first component is decomposed as

$$y = y' \cdot 2^{(p_A \circ p(\|\vec{x}\|)) \cdot p(\|\vec{x}\|)} + y'' \qquad \text{with } y'' < 2^{(p_A \circ p(\|\vec{x}\|) \cdot p(\|\vec{x}\|))}.$$

Consider the word $w' = 0 \ldots 0 \cdot \beta(y') \in \{0,1\}^*$ of length $|w'| = p(\|\vec{x}\|)$, where $\beta(y')$ denotes the usual binary expansion of the number y'. Thus, w' corresponds to an initial part of the binary expansion of y. This word is now interpreted as a sequence of oracle answers in the course of working of $\mathcal{M}$ on input $\vec{x}$. More precisely, for the lth bit b_l of w' let $b_l = 0$ mean the answer "yes" and $b_l = 1$ mean the answer "no" to the lth oracle query in the related path of working of $\mathcal{M}$ on input $\vec{x}$. The corresponding lth subword of length $p_A \circ p(\|\vec{x}\|)$ of the word $w'' = 0 \ldots 0 \cdot \beta(y'')$ with $|w''| = p_A \circ p(\|\vec{x}\|) \cdot p(\|\vec{x}\|)$ provides a potential witness y_l with $\|y_l\| \leq p_A \circ p(\|\vec{x}\|)$, $1 \leq l \leq p(\|\vec{x}\|))$, such that $h(y_l, z_l) = b_l$ for the lth oracle query z_l.

More precisely, on input $(y,\vec{x})$ let $\mathcal{M}'$ simulate the behavior of $\mathcal{M}$ for at most $p(\|\vec{x}\|)$ steps in such a way that the lth oracle query is always answered according to b_l and, moreover, it is checked whether $h(y_l, z_l) = b_l$, for the related part y_l of y'' and the current oracle query z_l. If $h(y_l, z_l) \neq b_l$, for some first such l, let $\mathcal{M}'$ enter a cycle of working (without halting, the result remains undefined); otherwise let it continue the simulation. If a stop configuration of $\mathcal{M}$ is reached (thus,

$h(y_l, z_l) = b_l$ for all oracle steps l performed so far), let $\mathcal{M}'$ output the result of machine $\mathcal{M}$ corresponding to the simulated path (controlled by the oracle answers according to the sequence of the bits b_l).

Since $\mathcal{M}$ computes the total function f and the zero bit $b_l = 0$ corresponds to the oracle answer "yes", it is ensured that the minimal value of y, for which the input $(y, \vec{x})$ yields a result of $\mathcal{M}'$, corresponds to a sequence of correct oracle answers in the related path of $\mathcal{M}$ with respect to the oracle set A. Thus, $\mathcal{M}'$ computes a 1-polynomial function $g \in \mathrm{FP}^{\Sigma_k^p}$ such that $f = \boldsymbol{\phi}'(g)$. This completes the proof of Lemma 8.2. □

If, in the above proof, the function f is allowed to be partial, then for $\vec{x} \notin \mathrm{dom}(f)$ there could be a (minimal) value y for which the input $(y, \vec{x})$ gives a result of $\mathcal{M}'$ (which corresponds to an incorrect sequence of oracle answers). Hence we cannot conclude that $f = \boldsymbol{\phi}'(g)$ in this case. If both the oracle set A and its complement $\overline{A}$ belong to Σ_{k+1}^p, however, one can even show that $f \in \mathrm{FNP}^{\Sigma_k^p}$. This means

Lemma 8.3. *For every $k \in \mathbb{N}$, $\mathrm{FP}^{\Sigma_{k+1}^p \cap \Pi_{k+1}^p} \subseteq \mathrm{FNP}^{\Sigma_k^p}$.*

To prove this, we show that for every (partial) function $f \in \mathrm{FP}^{\Sigma_{k+1}^p \cap \Pi_{k+1}^p}$, there exists a 1-polynomial $g \in \mathrm{FP}^{\Sigma_k^p}$ such that $f = \boldsymbol{\phi}'(g)$ and, moreover, $g(y_1, \vec{x}) = g(y_2, \vec{x})$ whenever $g(y_1, \vec{x})\downarrow$ and $g(y_2, \vec{x})\downarrow$.

Indeed, employing the framework and denotations from the proof of Lemma 8.2, we have for the complement $\overline{A} = \mathbb{N} \setminus A$ that

$$\overline{A} = \{ z \mid \exists^{p_A(\|z\|)} y \; \overline{h}(y, z) = 0 \}$$

with a 1-polynomial binary function $\overline{h} \in \mathrm{FP}^{\Sigma_k^p}$, for which $\overline{h}(y, z)\downarrow$ whenever $\|y\| \leq p_A(\|z\|)$, where the polynomial p_A is chosen to be applicable in the representation of both A and $\overline{A}$.

Now the behavior of the TM $\mathcal{M}'$ is modified such that in case that $b_l = 1$ the corresponding part y_l of y'' has to witness that $z_l \in \overline{A}$, *i.e.*, $\overline{h}(y_l, z_l) = 0$ with z_l corresponding to the lth oracle query of machine $\mathcal{M}$ on an input $(y, \vec{x})$. This ensures that all results obtained by $\mathcal{M}'$ are equal to $f(\vec{x})$ and that there is a number y such that $\mathcal{M}'$ halts with a result on input $(y, \vec{x})$ if $f(\vec{x})\downarrow$. On the other hand, for all inputs $(y, \vec{x})$ with $\vec{x} \notin \mathrm{dom}(f)$, $\mathcal{M}'$ cannot halt with a result, since all oracle queries have to be answered correctly then, and the simulation follows correctly the work of $\mathcal{M}$. Hence we have $f = \boldsymbol{\phi}'(g)$ for the 1-polynomial function $g \in \mathrm{FP}^{\Sigma_k^p}$ computed by $\mathcal{M}'$, and $g(y_1, \vec{x}) = g(y_2, \vec{x})$ whenever $g(y_1, \vec{x})\downarrow$ and $g(y_2, \vec{x})\downarrow$. This means $f \in \mathrm{FNP}^{\Sigma_k^p}$. □

We shall return to questions and problems around Lemmas 8.2 and 8.3 in Section 10. By the present section, the levels $\mathrm{FLev}'_{\boldsymbol{\omega}'}(k)$ for $\boldsymbol{\omega}' \in \{\overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}', \mathbf{min}', \mathbf{max}'\}$ have completely been characterized within the polynomial hierarchy. It has turned out that the behavior of the polytime variants of operators is not straightly analogous to the relationships between their computation theoretic counterparts in the arithmetical hierarchy. In particular, the question whether the polytime levels establish a proper hierarchy corresponds to the question whether the polynomial hierarchy does not collapse. Hence it is related to the open P *vs.* NP problem.

9. On the polytime levels of further operators

First we want to deal again with the polytime version of the classical $\boldsymbol{\mu}$-operator. Surprisingly, its levels will turn out to be equal to those of the previously discussed operators.

By Theorem 8.1, the levels of the operator $\overline{\boldsymbol{\mu}}'$ coincide with the classes $\mathrm{F}\Delta^p_{k+1}$ defined by means of (deterministic) POTMs. Hence they are closed under totalization in the following sense.

Let the operator of *totalization* $\boldsymbol{\tau}$ assign, to any function $f \in \mathrm{FAll}$, the function $\boldsymbol{\tau}(f) = \overline{f}$ of the same arity, which is defined by

$$\overline{f}(\vec{x}) = \begin{cases} f(\vec{x}) + 1 & \text{if } f(\vec{x}) \downarrow, \\ 0 & \text{if } f(\vec{x}) \uparrow. \end{cases}$$

Quite naturally, we say that a function class $\mathrm{FC} \subseteq \mathrm{FAll}$ is closed under totalization if $\boldsymbol{\tau}(f) \in \mathrm{FC}$ whenever $f \in \mathrm{FC}$. For example, all complexity classes of functions which are defined by means of deterministic TMs with the usual constructible time-bounds, as well as unions of such classes, are closed under totalization. In particular, the classes $\mathrm{F}\Delta^p_{k+1} = \mathrm{FLev}'(k)$ have this property.

The following Lemma shows that closedness under totalization means that the functions of the class under consideration have domains which are decidable within this class.

Lemma 9.1. *Let* $\mathrm{FP} \subseteq \mathrm{FC} \subseteq \mathrm{FAll}$ *for a function class* FC *which is closed under composition. Then* FC *is closed under totalization iff* $\chi_{\mathrm{dom}(f)} \in \mathrm{FC}$ *for all functions* $f \in \mathrm{FC}$.

Indeed, if FC is closed under totalization, for any $f \in \mathrm{FC}$ we have $\chi_{\mathrm{dom}(f)} = h \circ \boldsymbol{\tau}(f) \in \mathrm{FC}$, with the polytime function

$$h(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$

Conversely, it holds

$$\boldsymbol{\tau}(f)(\vec{x}) = \begin{cases} 0 & \text{if } \chi_{\mathrm{dom}(f)}(\vec{x}) = 0, \\ f(\vec{x}) + 1 & \text{if } \chi_{\mathrm{dom}(f)}(\vec{x}) = 1, \end{cases}$$

hence $\boldsymbol{\tau}(f) \in \mathrm{Clos}_{\{\circ\}}(\mathrm{FP} \cup \{f\}) \subseteq \mathrm{FC}$ under the supposition of the lemma. □

Now it can be shown that un-nested applications of the operators $\overline{\boldsymbol{\mu}}'$ and $\boldsymbol{\mu}'$ to functions of certain classes yield closely related results.

Lemma 9.2. *If* FP $\subseteq$ FC $\subseteq$ FAll *and the function class* FC *is closed under totalization and composition, then*

$$\mathrm{Clos}_{\{\circ\}}\,(\,\overline{\boldsymbol{\mu}}'(\mathrm{FC})\cup\mathrm{FC}\,)=\mathrm{Clos}_{\{\circ\}}\,(\,\boldsymbol{\mu}'(\mathrm{FC})\cup\mathrm{FC}\,).$$

If $f=\overline{\boldsymbol{\mu}}'(g)$ for an $(m+1)$-ary function g such that $g(y,\vec{x})\uparrow$ whenever $\|y\|>p(\|\vec{x}\|)$, with a polynomial p, we put

$$g'(y,\vec{x})\simeq\begin{cases}0 & \text{if } \|y\|\le p(\|\vec{x}\|) \text{ and } g(y,\vec{x})=0\,,\\ 1 & \text{if } \|y\|\le p(\|\vec{x}\|) \text{ and } (\,g(y,\vec{x})\uparrow \ \text{ or } g(y,\vec{x})>0\,)\,,\\ \uparrow & \text{otherwise .}\end{cases}$$

Then $g'=h\circ(\boldsymbol{\tau}(g),\varkappa_1^{m+1},\ldots,\varkappa_{m+1}^{m+1})$ with a suitable function $h\in$ FP, and we have $f=\boldsymbol{\mu}'(g')$. Under the supposition of the lemma it follows $g'\in$ FC, hence $\overline{\boldsymbol{\mu}}'(\mathrm{FC})\subseteq\boldsymbol{\mu}'(\mathrm{FC})$, what implies $\mathrm{Clos}_{\{\circ\}}\,(\,\overline{\boldsymbol{\mu}}'(\mathrm{FC})\cup\mathrm{FC}\,)\subseteq\mathrm{Clos}_{\{\circ\}}\,(\,\boldsymbol{\mu}'(\mathrm{FC})\cup\mathrm{FC}\,)$.

To show the converse inclusion, let $f=\boldsymbol{\mu}'(g)$ and $g(y,\vec{x})\uparrow$ whenever $\|y\|>p(\|\vec{x}\|)$, with a polynomial p. Now we put

$$g'(y,\vec{x})\simeq\begin{cases}g(y,\vec{x}) & \text{if } \|y\|\le p(\|\vec{x}\|) \text{ and } g(y,\vec{x})\downarrow\,,\\ 0 & \text{if } \|y\|\le p(\|\vec{x}\|) \text{ and } g(y,\vec{x})\uparrow\,,\\ \uparrow & \text{otherwise .}\end{cases}$$

Then $g'\in$ FC and we have

$$f(\vec{x})\simeq\begin{cases}\overline{\boldsymbol{\mu}}'(g')(\vec{x}) & \text{if } \overline{\boldsymbol{\mu}}'(g)(\vec{x})\downarrow \text{ and } \overline{\boldsymbol{\mu}}'(g')(\vec{x})\downarrow \ \text{ and } \overline{\boldsymbol{\mu}}'(g)(\vec{x})=\overline{\boldsymbol{\mu}}'(g')(\vec{x})\,,\\ \uparrow & \text{otherwise .}\end{cases}$$

Thus, $f\in\mathrm{Clos}_{\{\circ\}}\,(\,\overline{\boldsymbol{\mu}}'(\mathrm{FC})\cup\mathrm{FC}\,)$. □

By induction on k, from Lemma 9.2 and Theorem 8.1 we immediately obtain

Corollary 9.1. $\mathrm{FLev}'_{\boldsymbol{\mu}'}(k)=\mathrm{FLev}'(k)$ *for all* $k\in\mathbb{N}$. □

Next we want to deal with the remaining operators $\sharp'$ and $\mathbf{sum}'$. It is easily seen that they yield the same levels. First, analogously to part (iii) of the proof of Lemma 5.1, one shows that $\sharp'(g)\in\mathbf{sum}'(\mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FP}\cup\{g\}\,))$, for any 1-polynomial function g.

On the other hand, let $f=\mathbf{sum}'(g)$ and $g(y,\vec{x})\uparrow$ whenever $\|y\|>p(\|\vec{x}\|)$ for a polynomial p. In addition, we suppose that $g\in$ FC for a function class FC which is closed under totalization. Then putting

$$g'(\langle y,z\rangle,\vec{x})\simeq\begin{cases}0 & \text{if } g(y,\vec{x})\downarrow \ \text{ and } z<g(y,\vec{x})\,,\\ \uparrow & \text{otherwise ,}\end{cases}$$

we get a 1-polynomial function $g'\in\mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FC}\cup\mathrm{FP}\,)$ such that $f=\sharp'(g')$. So it holds

Lemma 9.3. *For any function class* FC *which is closed under totalization and under composition and includes* FP*, we have* $\sharp'(\mathrm{FC}) = \mathbf{sum}'(\mathrm{FC})$. □

Moreover, we shall employ the following technical lemma.

Lemma 9.4. *Let a function class* FC *be closed under totalization and include* FP*. Then its closure,* $\mathrm{Clos}_{\{\circ\}}(\,\mathrm{FC}\,)$*, is closed under totalization, too.*

Indeed, if $f = g\circ(h_1, \ldots, h_l)$, then

$$\boldsymbol{\tau}(f)(\vec{x}) = \begin{cases} g(h_1(\vec{x}), \ldots, h_l(\vec{x})) + 1 & \text{if } h_i(\vec{x})\downarrow \text{ for all } i \in \{1, \ldots, l\} \\ & \text{and } g(h_1(\vec{x}), \ldots, h_l(\vec{x}))\downarrow\,, \\ 0 & \text{otherwise .} \end{cases}$$

It follows that $\boldsymbol{\tau}(f) \in \mathrm{Clos}_{\{\circ\}}(\,\{\boldsymbol{\tau}(g), \boldsymbol{\tau}(h_1), \ldots, \boldsymbol{\tau}(h_l)\} \cup \mathrm{FP}\,) \subseteq \mathrm{Clos}_{\{\circ\}}(\,\mathrm{FC}\,)$, and this yields the assertion. □

Proposition 9.1. $\mathrm{FLev}'_{\sharp'}(k) = \mathrm{FLev}'_{\mathbf{sum}'}(k)$ *for all* $k \in \mathbb{N}$.

Indeed, by induction on k and by means of Lemmas 9.3 and 9.4, one shows simultaneously that $\mathrm{FLev}'_{\sharp'}(k) = \mathrm{FLev}'_{\mathbf{sum}'}(k)$ and that this class is closed under totalization and composition and includes FP, for every $k \in \mathbb{N}$. □

Unfortunately, we did not succeed in localizing the levels of $\sharp'$ and $\mathbf{sum}'$ within or compared to the polynomial hierarchy. It is even open whether they are contained in FPH. These problems correspond to the open questions concerning Valiant's counting class $\sharp$P well-known from complexity theory, *cf.*, *e.g.*, Chapter 9 in [7]. One easily sees that

$$\sharp\mathrm{P} = \sharp'(\mathrm{FP}) \subseteq \mathrm{FLev}'_{\sharp'}(1).$$

Thus, these unsolved hard problems concern already the first nonzero level of $\sharp'$ and $\mathbf{sum}'$, and it could even be that $\mathrm{FLev}_{\sharp'}(1)$ is not contained in FPH. The only related result, which we were able to prove so far within our framework, says that any level $\mathrm{FLev}'(k)$ is contained in the $\sharp'$-level of height $2k$.

Proposition 9.2. *For all* $k \in \mathbb{N}$, $\mathrm{FLev}'(k) \subseteq \mathrm{FLev}'_{\sharp'}(2k)$.

This follows by induction on k from the fact that the operator $\overline{\boldsymbol{\mu}}'$ can be expressed by a two-fold (nested) application of $\sharp'$, in composition with polytime functions.

More precisely, let $f = \overline{\boldsymbol{\mu}}'(g)$ for an $(m+1)$-ary function g satisfying $g(y, \vec{x})\uparrow$ whenever $\|y\| > p(\|\vec{x}\|)$, with a polynomial p. We put

$$g'(z, y, \vec{x}) \simeq \begin{cases} 0 & \text{if } z \leq y \text{ and } g(z, \vec{x}) = 0\,, \\ \uparrow & \text{otherwise ,} \end{cases}$$

and

$$h(y, \vec{x}) \simeq \begin{cases} 0 & \text{if } y < 2^{p(\|\vec{x}\|)} \text{ and } \sharp'(g')(y, \vec{x}) = 0\,, \\ \uparrow & \text{otherwise .} \end{cases}$$

Then g' and h are 1-polynomial functions, and we have $g' \in \mathrm{Clos}_{\{\circ\}}(\mathrm{FP} \cup \{g\})$ and $h \in \mathrm{Clos}_{\{\circ\}}(\mathrm{FP} \cup \sharp'(\mathrm{Clos}_{\{\circ\}}(\mathrm{FP} \cup \{g\})))$. Moreover,

$$\sharp'(h)(\vec{x}) = \begin{cases} 2^{p(\|\vec{x}\|)} & \text{if } \overline{\boldsymbol{\mu}}'(g)(\vec{x})\uparrow, \\ y < 2^{p(\|\vec{x}\|)} & \text{if } \overline{\boldsymbol{\mu}}'(g)(\vec{x}) = y. \end{cases}$$

Thus, if $g \in \mathrm{FLev}'_{\sharp'}(2k)$, then $f = \overline{\boldsymbol{\mu}}'(g) \in \mathrm{FLev}'_{\sharp'}(2k+2)$, and if $\mathrm{FLev}'(k) \subseteq \mathrm{FLev}'_{\sharp'}(2k)$, then $\mathrm{FLev}'(k+1) \subseteq \mathrm{FLev}'_{\sharp'}(2k+2)$. □

Propositions 7.1 and 9.2 immediately yield

Corollary 9.2. $\mathrm{FPH} \subseteq \bigcup_{k\in\mathbb{N}} \mathrm{FLev}'_{\sharp'}(k) = \mathrm{Clos}_{\{\circ,\sharp'\}}(\mathrm{FP})$. □

As we mentioned above, it is open whether the converse inclusion holds.

10. Equivalences to a collapse of the polynomial hierarchy

By Theorem 8.1 and Lemma 8.1, the operator levels $\mathrm{FLev}'(k)$, $k \in \mathbb{N}$, span the whole polynomial hierarchy of functions, which also characterizes the usual polynomial hierarchy of (classes of) sets. Thus, the polynomial hierarchy collapses iff the sequence of operator levels collapses to some level $\mathrm{FLev}'(k)$. The latter holds iff $\mathrm{FLev}'(k) = \mathrm{FLev}'(k+1)$, and this is the case iff $\mathrm{FLev}'(k)$ is closed under some (equivalently, under all) $\boldsymbol{\omega}' \in \{\boldsymbol{\mu}', \overline{\boldsymbol{\mu}}', \boldsymbol{\phi}', \boldsymbol{\varrho}', \mathbf{min}', \mathbf{max}'\}$. In this section, we establish several operator-related equivalent formulations of the question whether Σ^p_{k+1} is closed under complement. This problem is equivalent to $\Sigma^p_{k+1} = \Pi^p_{k+1}$ and to $\Sigma^p_{k+1} = \mathrm{FPH}$. For $k = 0$, it is just the famous NP *vs.* coNP problem.

We start with stating a basic chain of inclusions.

Proposition 10.1. *For all $k \in \mathbb{N}$,*

$$\begin{array}{ccccccccccc} \mathrm{FP}^{\Sigma^p_k} & \subseteq & \mathrm{FP}^{\Sigma^p_{k+1}\cap\Pi^p_{k+1}} & \subseteq & \mathrm{FNP}^{\Sigma^p_k} & \subseteq & \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma^p_k}) & \subseteq & \mathrm{FP}^{\Sigma^p_{k+1}} & \subseteq & \mathrm{FPH}. \\ \| & & & & \| & & & & \| & & \\ \mathrm{F}\Delta^p_{k+1} & & & & \mathrm{F}\Sigma^p_{k+1} & & & & \mathrm{F}\Delta^p_{k+2} & & \end{array}$$

The indicated equalities to the classes in the last line of the proposition hold by definition of the classes $\mathrm{F}\Delta^p_k$ and by Lemma 7.1, respectively. The first inclusion follows from $\Sigma^p_k \subseteq \Sigma^p_{k+1} \cap \Pi^p_{k+1}$, the second one is just Lemma 8.3, the third one holds by definition of the classes FNP^C, whereas the last one is trivial. Finally, $\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma^p_k}) \subseteq \mathrm{FP}^{\Sigma^p_{k+1}}$ was shown in the first part of the proof of Theorem 8.1. □

As a counterpart of the totalization introduced at the beginning of the previous section, now we define the operator of *partialization*, $\boldsymbol{\pi}$. It assigns, to any function $f \in \mathrm{FAll}$, the function $\boldsymbol{\pi}(f) = \widetilde{f}$ of the same arity, which is defined by

$$\widetilde{f}(\vec{x}) \simeq \begin{cases} f(\vec{x}) - 1 & \text{if } f(\vec{x})\downarrow \text{ and } f(\vec{x}) > 0, \\ \uparrow & \text{otherwise}. \end{cases}$$

Thus, $\boldsymbol{\pi}(f) = h_0 \circ f$, where the unary polytime function h_0 is defined by

$$h_0(x) \simeq \begin{cases} x-1 & \text{if } x > 0, \\ \uparrow & \text{if } x = 0. \end{cases}$$

For example, all function classes, which contain FP and are closed under composition, are closed under partialization, too. This holds for all classes occurring in Proposition 10.1, possibly except $\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$. On the other hand, all the classes occurring in Proposition 10.1, possibly except $\mathrm{FNP}^{\Sigma_k^p}$, are closed under totalization.

The variety of statements known as equivalent to $\Sigma_{k+1}^p = \Pi_{k+1}^p$ can now be enriched by some ones concerning equations between and closure properties of the above function classes.

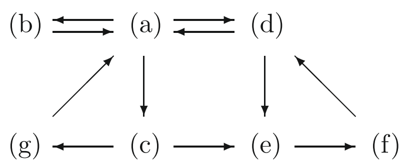
Theorem 10.1. *The following statements are equivalent:*

(a) $\Sigma_{k+1}^p = \Pi_{k+1}^p$;
(b) $\mathrm{FP}^{\Sigma_{k+1}^p \cap \Pi_{k+1}^p} = \mathrm{FNP}^{\Sigma_k^p}$;
(c) $\mathrm{FNP}^{\Sigma_k^p} = \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$;
(d) $\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p}) = \mathrm{FP}^{\Sigma_{k+1}^p}$;
(e) $\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$ *is closed under composition*;
(f) $\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$ *is closed under partialization*;
(g) $\mathrm{FNP}^{\Sigma_k^p}$ *is closed under totalization.*

Also, $\mathrm{FP}^{\Sigma_k^p} = \mathrm{FP}^{\Sigma_{k+1}^p \cap \Pi_{k+1}^p}$ *iff* $\Delta_{k+1}^p = \Sigma_{k+1}^p \cap \Pi_{k+1}^p$, *and from* $\Sigma_{k+1}^p = \Pi_{k+1}^p$ *it follows that* $\mathrm{FP}^{\Sigma_{k+1}^p \cap \Pi_{k+1}^p} = \mathrm{FPH}$ *and conversely.*

The structure of our proof of the equivalence of the statements (a)–(g) can be sketched by the following scheme.

Scheme of implications
shown in the proof of Theorem 10.1



Obviously, $\Sigma_{k+1}^p = \Pi_{k+1}^p$ implies $\mathrm{FP}^{\Sigma_{k+1}^p \cap \Pi_{k+1}^p} = \mathrm{FP}^{\Sigma_{k+1}^p}$, *i.e.*, from (a) follow (b), (c) and (d). Moreover, it is well-known that (a) implies that the polynomial hierarchy collapses to Σ_{k+1}^p, *i.e.*, $\Sigma_{k+1}^p = \mathrm{PH}$, *cf.* [7]. Then it follows $\mathrm{F}\Sigma_{k+1}^p = \mathrm{FPH}$. So we have already shown the main direction of the theorem's very last statement, the converse implication follows from the equivalence of (a)–(g).

Since $\mathrm{FNP}^{\Sigma_k^p}$ and $\mathrm{FP}^{\Sigma_{k+1}^p}$ are closed under composition, both (c) and (d) imply (e). (e)⇒(f) follows, since $h_0 \in \mathrm{FP} \subseteq \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$, and (c)⇒(g) holds, since $\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$ is closed under totalization, as one easily sees.

To show (d)⇒(a), assume that $\mathrm{FP}^{\Sigma^p_{k+1}} = \{\boldsymbol{\phi}'(g) : g \in \mathrm{FP}^{\Sigma^p_k}\}$. If $A \in \Sigma^p_{k+1}$, we have $\chi^0_A \in \mathrm{FNP}^{\Sigma^p_k} \subseteq \mathrm{FP}^{\Sigma^p_{k+1}}$. Due to the abilities of POTMs, it follows $\chi^0_{\overline{A}} \in \mathrm{FP}^{\Sigma^p_{k+1}}$. Hence, by supposition (d), there is a 1-polynomial $\overline{g} \in \mathrm{FP}^{\Sigma^p_k}$ with $\chi^0_{\overline{A}} = \boldsymbol{\phi}'(\overline{g})$. One can easily ensure that $\overline{g}(y, \vec{x}) = 0$ if $\overline{g}(y, \vec{x})\downarrow$. Thus, $\chi^0_{\overline{A}} \in \mathrm{FNP}^{\Sigma^p_k}$ and $\overline{A} \in \Sigma^p_{k+1}$, *i.e.*, $A \in \Pi^p_{k+1}$. So we have shown $\Sigma^p_{k+1} \subseteq \Pi^p_{k+1}$, and this implies the equality (a).

Now suppose (b). For any $A \in \Sigma^p_{k+1}$, we have $\chi^0_A \in \mathrm{FNP}^{\Sigma^p_k} = \mathrm{FP}^{\Sigma^p_{k+1} \cap \Pi^p_{k+1}}$. Again by the abilities of POTMs, it follows $\chi^0_{\overline{A}} \in \mathrm{FP}^{\Sigma^p_{k+1} \cap \Pi^p_{k+1}} = \mathrm{FNP}^{\Sigma^p_k}$ and $\overline{A} \in \Sigma^p_{k+1}$. Thus, we have (b)⇒(a).

Let (f) hold, *i.e.*, $\boldsymbol{\pi}(g) \in \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma^p_k})$ for any $g \in \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma^p_k})$. Given some $g \in \mathrm{FP}^{\Sigma^p_{k+1}}$, its totalization $\boldsymbol{\tau}(g)$ belongs to $\mathrm{FP}^{\Sigma^p_{k+1}}$ too. By Lemma 8.2, $\boldsymbol{\tau}(g) \in \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma^p_k})$, and (f) implies that $g = \boldsymbol{\pi} \circ \boldsymbol{\tau}(g) \in \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma^p_k})$. So we have shown $\mathrm{FP}^{\Sigma^p_{k+1}} \subseteq \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma^p_k})$; this means (d).

Finally, let $A \in \Sigma^p_{k+1} = \mathrm{NP}^{\Sigma^p_k}$, *i.e.*, $\chi^0_A \in \mathrm{FNP}^{\Sigma^p_k}$. If (g) holds, then it follows $\chi_A \in \mathrm{FNP}^{\Sigma^p_k}$, hence $\chi^0_{\overline{A}} \in \mathrm{FNP}^{\Sigma^p_k}$. So we have shown that (g) implies $\Sigma^p_{k+1} \subseteq \Pi^p_{k+1}$, and this yields (a). The proofs of the implications indicated above are complete.

If $\Delta^p_{k+1} = \Sigma^p_{k+1} \cap \Pi^p_{k+1}$, then $\mathrm{FP}^{\Sigma^p_{k+1} \cap \Pi^p_{k+1}} = \mathrm{FP}^{\mathrm{P}^{\Sigma^p_k}} = \mathrm{FP}^{\Sigma^p_k}$. Conversely, $\mathrm{FP}^{\Sigma^p_k} = \mathrm{FP}^{\Sigma^p_{k+1} \cap \Pi^p_{k+1}}$ and $A \in \Sigma^p_{k+1} \cap \Pi^p_{k+1}$ implies $\chi_A \in \mathrm{FP}^{\Sigma^p_{k+1} \cap \Pi^p_{k+1}} = \mathrm{FP}^{\Sigma^p_k}$, hence $A \in \mathrm{P}^{\Sigma^p_k} = \Delta^p_{k+1}$. This completes the proof of the theorem. □

The most interesting special case of Proposition 10.1 and Theorem 10.1 is obtained for $k = 0$ and, accordingly, $\Sigma^p_1 = \mathrm{NP}$, $\Pi^p_1 = \mathrm{coNP}$, $\mathrm{F}\Delta^p_1 = \mathrm{FP}^{\mathrm{P}} = \mathrm{FP}$, $\mathrm{FNP}^{\Sigma^p_0} = \mathrm{FNP}$, and $\Delta^p_1 = \mathrm{P}$. We re-formulate these results as a corollary. It should be mentioned that the implication $\mathrm{NP} = \mathrm{coNP} \Rightarrow \mathrm{FNP} = \mathrm{F}\Delta^p_2$ was essentially known from [26], Theorem 11.1. Moreover, there are known a lot of further equations or conditions equivalent to $\mathrm{NP} = \mathrm{coNP}$.

Corollary 10.1. *We have* $\mathrm{FP} \subseteq \mathrm{FP}^{\mathrm{NP} \cap \mathrm{coNP}} \subseteq \mathrm{FNP} \subseteq \boldsymbol{\phi}'(\mathrm{FP}) \subseteq \mathrm{F}\Delta^p_2 \subseteq \mathrm{FPH}$, *and the following statements are equivalent:*

(a) $\mathrm{NP} = \mathrm{coNP}$;
(b) $\mathrm{FP}^{\mathrm{NP} \cap \mathrm{coNP}} = \mathrm{FNP}$;
(c) $\mathrm{FNP} = \boldsymbol{\phi}'(\mathrm{FP})$;
(d) $\boldsymbol{\phi}'(\mathrm{FP}) = \mathrm{F}\Delta^p_2$;
(e) $\boldsymbol{\phi}'(\mathrm{FP})$ *is closed under composition*;
(f) $\boldsymbol{\phi}'(\mathrm{FP})$ *is closed under partialization*;
(g) FNP *is closed under totalization.*

Moreover, $\mathrm{FP} = \mathrm{FP}^{\mathrm{NP}\cap\mathrm{coNP}}$ *iff* $\mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}$*, and from* $\mathrm{NP} = \mathrm{coNP}$ *it follows that* $\mathrm{FP}^{\mathrm{NP}\cap\mathrm{coNP}} = \mathrm{FPH}$ *and conversely.*

Theorem 7.1 and Corollaries 7.1 and 10.1 emphasize once more that questions around P *vs.* NP can be expressed quite independently on the notion of nondeterminism, *e.g.*, by closure properties of the basic class FP with respect to our variants of classical operators. Thus, it is stressed that the hardness of P *vs.* NP questions is caused by an insufficient knowledge of the class FP of polytime functions.

Remembering Propositions 6.1 and 6.2 and Corollary 9.1, one might ask whether the operators $\boldsymbol{\mu}'$, $\overline{\boldsymbol{\mu}}'$, $\boldsymbol{\varrho}'$, $\mathbf{min}'$ and $\mathbf{max}'$ would yield results similar to Theorem 10.1 and Corollary 10.1 if they are taken instead of $\boldsymbol{\phi}'$. This indeed holds for $\mathbf{min}'$ and $\mathbf{max}'$, whereas $\boldsymbol{\mu}'$, $\overline{\boldsymbol{\mu}}'$ and $\boldsymbol{\varrho}'$ do not seem to allow such analogues.

11. A hierarchy between $\mathrm{FNP}^{\Sigma_k^p}$ and $\phi'(\mathrm{FP}^{\Sigma_k^p})$

The previous section demonstrated that the polytime version of the first value operator is well suited to express statements equivalent to a collapse of the polynomial hierarchy. Now this operator will be used to establish a hierarchy of function classes between $\mathrm{FNP}^{\Sigma_k^p}$ and $\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$, for any $k \in \mathbb{N}$. This corresponds to the generalized Boolean hierarchy over $\Sigma_{k+1}^p = \mathrm{NP}^{\Sigma_k^p}$ which leads from Σ_{k+1}^p to $\Delta_{k+2}^p = \mathrm{P}^{\Sigma_{k+1}^p}$ and was explored in detail for $k = 0$. There one first gets the Boolean hierarchy over $\mathrm{NP} = \Sigma_1^p$ that corresponds to the Boolean closure of NP but can be extended to a hierarchy exhausting the whole class Δ_2^p, *cf.* [4,5,31,32].

The basic technique of the construction of the Boolean closure of a "ring of sets" goes back to Hausdorff [11]. His difference hierarchy was transferred to computability theory by Ershov [9]. For a unified treatment of both these hierarchies and of a counterpart in constructive analysis, the reader is referred to [12]. Epstein, Haas and Kramer [8] introduced (the original version of) the first value operator just in order to investigate Ershov's hierarchy. In [13] we applied related techniques to establish hierarchies of function classes which, in a certain sense, generalize both the Hausdorff hierarchy and the Ershov hierarchy, as well as their counterpart in constructive analysis.

In the present setting, a hierarchy below $\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$ can be created by restricting the number of the changes of values of 1-polynomial polytime functions from $\mathrm{FP}^{\Sigma_k^p}$ to which the first value operator is applied. More precisely, for an $(m+1)$-ary 1-polynomial function g, let

$$\begin{aligned}\alpha_g(\vec{x}) \quad = \quad & \mathrm{card}\{(z_1, z_2) : z_1 < z_2,\ g(z_1, \vec{x})\!\downarrow,\ g(z_2, \vec{x})\!\downarrow,\ g(z_1, \vec{x}) \neq g(z_2, \vec{x}), \\ & \qquad \text{and } g(z, \vec{x})\!\uparrow \text{ for all } z \text{ with } z_1 < z < z_2\}.\end{aligned}$$

Notice that, due to the 1-polynomiality of g, $\alpha_g(\vec{x})$ is always a natural number.

For any unary total function $\beta : \mathbb{N} \longrightarrow \mathbb{N}_+$, the related Φ_k-*class* is defined as

$$\Phi_{k,\beta(n)} = \{\boldsymbol{\phi}'(g) \colon g \in \mathrm{FP}^{\Sigma_k^p} \text{ is 1-polynomial, and } \alpha_g(\vec{x}) < \beta(\|\vec{x}\|) \text{ for all } \vec{x} \in \mathbb{N}^m\}.$$

One immediately sees that

$$\begin{array}{rll} \mathrm{FNP}^{\Sigma_k^p} = \Phi_{k,1} \subseteq \Phi_{k,\beta(n)} \subseteq \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p}) & \text{for all } \beta : \mathbb{N} \longrightarrow \mathbb{N}_+, & \text{and} \\ \Phi_{k,\beta_1(n)} \subseteq \Phi_{k,\beta_2(n)} & \text{if } \beta_1(n) \leq \beta_2(n) \text{ for (almost) all } n \in \mathbb{N}. & \end{array}$$

Moreover, if $g(y, \vec{x})\uparrow$ whenever $\|y\| > p(\|\vec{x}\|)$ for some $g \in \mathrm{FP}^{\Sigma_k^p}$, then $\boldsymbol{\phi}'(g) \in \Phi_{k,2^{p(n)}}$. Thus, $\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$ is obtained as the union over all classes $\Phi_{k,\beta(n)}$ for $\beta(n) \leq 2^{p(n)}$ with polynomials p, *i.e.*,

$$\boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p}) = \bigcup\{\Phi_{k,2^{p(n)}} : p \text{ is a polynomial}\} = \bigcup\{\Phi_{k,2^{(n^d)}} : d \in \mathbb{N}\}.$$

From any strict inclusion $\Phi_{k,\beta_1(n)} \subset \Phi_{k,\beta_2(n)}$, for unary total functions β_1 and β_2, it would follow that $\mathrm{FNP}^{\Sigma_k^p} \subset \boldsymbol{\phi}'(\mathrm{FP}^{\Sigma_k^p})$, *i.e.*, $\Sigma_{k+1}^p \neq \Pi_{k+1}^p$ by Theorem 10.1.

Now we are going to show a close relationship between the hierarchy of Φ_k-classes and the Boolean hierarchy over $\mathrm{NP}^{\Sigma_k^p}$. To this purpose, we transfer the generalized Boolean hierarchy over NP, *cf.* [1,31,32], both to sets of tuples of natural numbers and to higher levels of the polynomial hierarchy in the following straightforward way:
For any function $\beta : \mathbb{N} \longrightarrow \mathbb{N}_+$, let the class $\mathrm{BH}_{k,\beta(n)}$ consist of all sets $A \subseteq \mathbb{N}^m$, $m \in \mathbb{N}_+$, for which there is a set $B \in \mathrm{NP}^{\Sigma_k^p}$, $B \subseteq \mathbb{N}^{m+1}$, such that

$$(*) \quad \left\{ \begin{array}{l} (y, \vec{x}) \in B \text{ implies } (y', \vec{x}) \in B \text{ whenever } 1 \leq y' \leq y, \quad \text{and} \\ \vec{x} \in A \ \text{ iff } \ \max\{y : 1 \leq y \leq \beta(\|\vec{x}\|) \text{ and } (y, \vec{x}) \in B\} \equiv 1 \pmod{2}, \end{array} \right.$$

where, of course, $\max \emptyset \not\equiv 1 \pmod{2}$. Putting

$$B_y = \left\{ \begin{array}{ll} \{\vec{x} : (y, \vec{x}) \in B\} & \text{if } 1 \leq y \leq \beta(\|\vec{x}\|), \\ \emptyset & \text{if } y > \beta(\|\vec{x}\|), \end{array} \right.$$

condition $(*)$ can also be expressed by

$$(+) \quad \left\{ \begin{array}{l} B_1 \supseteq B_2 \supseteq B_3 \supseteq \ \ldots \ , \text{ and} \\ A = \bigcup_{y \in \mathbb{N}} (B_{2y+1} \setminus B_{2y+2}). \end{array} \right.$$

This representation is closely related to Hausdorff's difference hierarchy: A is the result of the difference chain of the decreasing sequence $(B_y)_{y \in \mathbb{N}_+}$.

By standard techniques, one shows:

$$
\begin{aligned}
\Sigma^p_{k+1} = \mathrm{NP}^{\Sigma^p_k} \;=\; \mathrm{BH}_{k,1} \;\subseteq\; \mathrm{BH}_{k,\beta(n)} \qquad & \text{for all } \beta : \mathbb{N} \longrightarrow \mathbb{N}_+\,,\\
\mathrm{BH}_{k,\beta(n)} \;\subseteq\; \mathrm{P}^{\Sigma^p_{k+1}} = \Delta^p_{k+2} \qquad & \text{if } \beta \text{ is polynomial-time computable, and}\\
\mathrm{BH}_{k,\beta_1(n)} \;\subseteq\; \mathrm{BH}_{k,\beta_2(n)} \qquad & \text{if } \beta_1(n) \le \beta_2(n) \text{ for (almost) all } n \in \mathbb{N}.
\end{aligned}
$$

The standard Boolean hierarchy over NP is obtained in this way for $k = 0$ and constant functions $\beta(n) = l \in \mathbb{N}$:

$$
\mathrm{NP} = \mathrm{BH}_{1,1} \subseteq \mathrm{BH}_{1,2} \subseteq \;\ldots\; \subseteq \mathrm{BH}_{1,l} \subseteq \;\ldots\; \subseteq \Delta^p_2,
$$

and the class $\bigcup\{\mathrm{BH}_{1,l} : l \in \mathbb{N}_+\}$ is just the Boolean closure of NP.

In the following theorem, the bounding functions β are supposed to be both polytime and *polynomially bounded*. The latter means that there is a polynomial p such that $\beta(n) \le p(n)$ for all $n \in \mathbb{N}$.

Theorem 11.1. *Let $\beta : \mathbb{N} \longrightarrow \mathbb{N}_+$, $\beta \in \mathrm{FP}$ and β polynomially bounded. Then for any $A \subseteq \mathbb{N}^m$, $m \in \mathbb{N}_+$, we have:*

$A \in \mathrm{BH}_{k,\beta(n)}$ *iff* *there is a function $f \in \Phi_{k,\beta(n)}$ such that $\vec{x} \in A$ iff $f(\vec{x}) = 1$.*

To prove the implication "$\Rightarrow$", let $A \in \mathrm{BH}_{k,\beta(n)}$ be characterized by a set $B \in \mathrm{NP}^{\Sigma^p_k}$ according to $(*)$, where the function β fulfills the required suppositions. Since $\chi^0_B \in \mathrm{FNP}^{\Sigma^p_k}$, there is a 1-polynomial function $g \in \mathrm{FP}^{\Sigma^p_k}$ such that for all $\vec{x} \in \mathbb{N}^m$, $z_1, z_2, y \in \mathbb{N}$:

if $g(z_1, y, \vec{x})\!\downarrow$ and $g(z_2, y, \vec{x})\!\downarrow$, then $g(z_1, y, \vec{x}) = g(z_2, y, \vec{x})$,
and it holds $(y, \vec{x}) \in B$ iff there is a $z \in \mathbb{N}$ with $g(z, y, \vec{x}) = 0$.

Without loss of generality, we suppose that $\mathrm{ran}(g) \subseteq \{0, 1\}$ and always $g(z, 0, \vec{x})\!\uparrow$. Let p be a polynomial witnessing that the function g is 1-polynomial. Now we put

$$
h(y \cdot 2^{p(\|(\beta(\|\vec{x}\|),\vec{x})\|)} + z, \vec{x}) \simeq
\begin{cases}
1 & \text{if } \|y\| \le \beta(\|\vec{x}\|),\ \|z\| \le p(\|\langle y, \vec{x}\rangle\|),\\
 & \quad g(z, \beta(\|\vec{x}\|) - y, \vec{x}) = 0 \text{ and } \beta(\|\vec{x}\|) - y \equiv 1 \pmod{2},\\
0 & \text{if } \|y\| \le \beta(\|\vec{x}\|),\ \|z\| \le p(\|\langle y, \vec{x}\rangle\|),\\
 & \quad g(z, \beta(\|\vec{x}\|) - y, \vec{x}) = 0 \text{ and } \beta(\|\vec{x}\|) - y \equiv 0 \pmod{2},\\
\uparrow & \text{otherwise.}
\end{cases}
$$

This $(m+1)$-ary function h is 1-polynomial, belongs to $\mathrm{FP}^{\Sigma^p_k}$ like g, and it holds: $\vec{x} \in A$ iff $\boldsymbol{\phi}'(h)(\vec{x}) = 1$. Moreover, $\alpha_h(\vec{x}) < \beta(\|\vec{x}\|)$. Thus, $f = \boldsymbol{\phi}'(h)$ fulfills all the requirements from Theorem 11.1.

To prove "$\Leftarrow$", let be given $f \in \Phi_{k,\beta(n)}$ and $A \subseteq \mathbb{N}^m$ such that $\vec{x} \in A$ iff $f(\vec{x}) = 1$. Let $f = \boldsymbol{\phi}'(g)$, where $g \in \mathrm{FP}^{\Sigma_k^p}$, g is 1-polynomial, and $\alpha_g(\vec{x}) < \beta(\|\vec{x}\|)$. We can suppose that $\mathrm{ran}(f), \mathrm{ran}(g) \subseteq \{0,1\}$.

For $1 \le \|y\| \le \beta(\|\vec{x}\|)$, we define $B \subseteq \mathbb{N}^{m+1}$ by

$$\begin{aligned}(y,\vec{x}) \in B \quad \text{iff} \quad & \text{there are } z_1 < z_2 < \cdots < z_y \text{ such that } g(z_i,\vec{x})\downarrow \\ & \text{whenever } 1 \le i \le y, \text{ and } g(z_i,\vec{x}) \ne g(z_{i+1},\vec{x}) \text{ for } 1 \le i < y, \\ & \text{and } ((y \text{ is odd and } g(z_1,\vec{x}) = 1) \text{ or } (y \text{ is even and } g(z_1,\vec{x}) = 0)).\end{aligned}$$

Then $B \in \mathrm{NP}^{\Sigma_k^p}$ and $(y,\vec{x}) \in B$ obviously implies $(y',\vec{x}) \in B$ whenever $1 \le y' \le y$.

Let $\vec{x} \in A$. Then $\boldsymbol{\phi}'(g)(\vec{x}) = 1 = g(z_0,\vec{x})$, where $z_0 = \min\{z : g(z,\vec{x})\downarrow\}$. Thus, $\vec{x} \in B_1$, and $\vec{x} \in B_2$ iff there are $z_1 < z_2$ such that $g(z_1,\vec{x}) = 0$ and $g(z_1,\vec{x}) = 1$. In the latter case, we have $z_0 < z_1$ and $\vec{x} \in B_3$. Then $\vec{x} \in B_4$ iff there are $z_1 < z_2 < z_3 < z_4$ such that $g(z_1,\vec{x}) = g(z_3,\vec{x}) = 0$ and $g(z_2,\vec{x}) = g(z_4,\vec{x}) = 1$. This holds iff $\vec{x} \in B_5$, and so on. It is seen that $\max\{y : (y,\vec{x}) \in B\}$ is an odd number $\le \beta(\|\vec{x}\|)$.

Conversely, if $\max\{y : (y,\vec{x}) \in B\}$ is an odd number, then it is $\le \beta(\|\vec{x}\|)$ and $\vec{x} \in A$. Thus, condition $(*)$ is fulfilled by the set B, *i.e.*, $A \in \mathrm{BH}_{k,\beta(n)}$. □

It should be noticed that, even if we could require that $\mathrm{ran}(f) \subseteq \{0,1\}$ for the function $f \in \Phi_{k,\beta(n)}$ according to Theorem 11.1, f is not necessarily total, *i.e.*, we could not conclude that $f = \chi_A$. For $\vec{x} \notin A$, we would only have $f(\vec{x}) = 0$ or $f(\vec{x})\uparrow$. From this it follows easily that $\chi_A \in \Phi_{k,\beta(n)+1}$ what, however, only implies that $A \in \mathrm{BH}_{k,\beta(n)+1}$. This situation is quite similar to those in the related settings where the first value operator is employed to characterize hierarchies of set classes, *cf.* [13].

Moreover, we would like to mention an interesting difference between the Boolean hierarchy and that of the Φ_k-classes of partial functions. For the generalized Boolean hierarchy over NP, in [31] it was proved that

$$\bigcup\{\mathrm{BH}_{1,\beta(n)} : \beta : \mathbb{N} \longrightarrow \mathbb{N}\} = \bigcup\{\mathrm{BH}_{1,2^{p(n)}} : p \text{ is a polynomial}\} = \Delta_2^p.$$

This corresponds to Lemma 8.2 which says that all the total functions from $\mathrm{F}\Delta_{k+2}^p$ belong to $\boldsymbol{\phi}'(\mathrm{F}\Delta_{k+1}^p) = \bigcup\{\Phi_{k,2^{p(n)}} : p \text{ is a polynomial}\}$. Thus, for $k = 0$ we have: all the total functions from $\mathrm{F}\Delta_2^p$ belong to $\boldsymbol{\phi}'(\mathrm{FP}) = \bigcup\{\Phi_{1,2^{p(n)}} : p \text{ is a polynomial}\}$. For the Φ_k-hierarchy it holds by Theorem 10.1:

$$\text{if } \bigcup\{\Phi_{k,2^{p(n)}} : p \text{ is a polynomial}\} = \mathrm{F}\Delta_{k+1}^p, \text{ then } \Sigma_{k+1}^p = \Pi_{k+1}^p.$$

Thus, probably there are functions in $f \in \mathrm{F}\Delta_{k+2}^p \setminus \bigcup\{\Phi_{k,2^{p(n)}} : p \text{ is a polynomial}\}$, but they have to be properly partial.

12. Concluding remarks

We have studied the nesting levels of some relatives of the classical $\boldsymbol{\mu}$-operator and their polytime counterparts. The computation theoretic variants span the stratification of the arithmetical hierarchy in different ways, whereas the polytime variants of most of the operators generate the polynomial hierarchy of functions. However, there seem to be only few analogies in the ways in which the nesting levels of the polytime operators and of the computation theoretic ones, respectively, are related to the stratifications of the hierarchies.

Proposition 3.1 characterizes the levels of odd heights, $\mathrm{FLev}(1+2k)$ for $k \in \mathbb{N}$. Its proof shows the strict inclusions $\mathrm{FLev}(1+2k) \subset \mathrm{FLev}(1+2k+1) \subset \mathrm{FLev}(1+2k+2)$. The levels of even heights, $\mathrm{FLev}(1+2k+1)$, are not characterized in more detail, however. This remains an open problem.

Also some further known function operators could be considered within the framework of this paper. Questions of the (un-nested) expressibility of one unary operator $\boldsymbol{\omega}_1$ by another one $\boldsymbol{\omega}_2$ would lead to study relations like $\leq_c$ defined by

$$\boldsymbol{\omega}_1 \leq_c \boldsymbol{\omega}_2 \quad \text{iff} \quad \boldsymbol{\omega}_1(g) \in \mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FPaPrim} \cup \{g\} \cup \boldsymbol{\omega}_2(\mathrm{Clos}_{\{\circ\}}\,(\,\mathrm{FPaPrim} \cup \{g\}\,)))$$
$$\text{for all } g \in \mathrm{dom}(\boldsymbol{\omega}_1).$$

In the case of polytime operators, one should define a relation $\leq_p$ accordingly by using the class FP instead of FPaPrim above. Lemmas 2.1 and 6.2 as well as Proposition 6.2 provide examples of such expressibilities between operators. Notice, however, that Lemma 5.1 and Proposition 5.1, as well as Lemmas 9.2 and 9.3, uses different techniques based on some knowledge about the nesting levels of the operators.

Quite naturally, questions about the strictness of the hierarchies of nesting levels of the polytime operators are closely related to the big unsolved problems of computational complexity, *cf.* Theorems 7.1, 8.1, 10.1 and Corollary 10.1. So we could perhaps present some new points of view for approaching these problems.

Finally, the previous Section 11 emphasizes the close connection between the first value operator and difference hierarchies of classes of sets or hierarchies of function classes related to them. In particular, the role of partial functions is stressed in this context. As we remarked, applications of the first value operator in its original form to functions from $\mathrm{F}\Sigma_1 = \mathrm{FCom}$ were studied in [8,13]. The computation theoretic analogues of the Φ_k hierarchies, however, would yield special hierarchies of function classes between $\mathrm{F}\Sigma_{1+k}$ and $\boldsymbol{\phi}(\mathrm{F}\Sigma_{1+k})$, for all $k \in \mathbb{N}$. These have not yet been dealt with in the present paper or somewhere else.

Acknowledgements. I want to thank the referees of earlier versions of this paper for their suggestions improving the style of presentation.

References

[1] S. Buss and L. Hay, On truth-table reducibility to SAT. *Inf. Comput.* **91** (1991) 86–102.

[2] J.L. Balcazar, J. Diaz and J. Gabarro, *Structural complexity I and II.* Springer-Verlag, Berlin (1990).

[3] S.J. Bellantoni and K.-H. Niggl, Ranking primitive recursions: the low Grzegorczyk classes revised. *SIAM Journal on Computing* **29** (1999) 401–415.

[4] J.-Y. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sawelson, K. Wagner and G. Wechsung, The Boolean hierarchy I: structural properties. *SIAM Journal on Computing* **17** (1988) 1232–1252.

[5] J.-Y. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sawelson, K. Wagner and G. Wechsung, The Boolean hierarchy II: applications. *SIAM Journal on Computing* **18** (1989) 95–111.

[6] S.B. Cooper, *Computability theory*. Chapman & Hall/CRC, Boca Raton (2004).

[7] D.-Z. Du and K.-I. Ko, *Theory of computational complexity*. Wiley-Interscience, New York (2000).

[8] R.L. Epstein, R. Haas and R.L. Kramer, *Hierarchies of sets and degrees below* $0'$. In: Logic Year (1979/80), Univ. of Connecticut, edited by M. Lerman, J.H. Schmerl, R.I. Soare. LN in Math 859. Springer Verlag, 32–48.

[9] Y.L. Ershov, A hierarchy of sets. I; II; III. Algebra i Logica **7** (1968) no. 1, 47–74; no. 4, 15–47; **9** (1970), no. 1, 34–51 (English translation by Plenum P.C.).

[10] M.R. Garey and D.S. Johnson, *Computers and intractability – a guide to the theory of NP–completeness*. W.H. Freeman, San Francisco (1979).

[11] F. Hausdorff, *Grundzüge der Mengenlehre*. W. de Gruyter & Co., Berlin and Leipzig (1914); Reprint: Chelsea P.C., New York (1949).

[12] A. Hemmerling, The Hausdorff-Ershov hierarchy in Euclidean spaces. *Arch. Math. Logic* **45** (2006) 323–350.

[13] A. Hemmerling, Hierarchies of function classes defined by the first-value operator. RAIRO - Theor. Inf. Appl. **42** (2008) 253–270.
Extended abstract in: Proc. of CCA'2004. *Electronic Notes in Theoretical Computer Science* **120** (2005) 59–72.

[14] J. Krajicek, *Bounded arithmetic, propositional logic, and complexity theory*. Cambridge Univ. Press (1995).

[15] M.W. Krentel, The complexity of optimization problems. *J. Comput. Syst. Sci.* **36** (1988) 490–509.

[16] A.I. Malcev, *Algorithmen und rekursive Funktionen*. Akademie-Verlag, Berlin (1974).

[17] P. Odifreddi, *Classical recursion theory*. North-Holland P.C., Amsterdam (1989).

[18] C.H. Papadimitriou, *Computational complexity*. Addison Wesley P.C., Reading (1994).

[19] C. Parsons, Hierarchies of primitive recursive functions. Zeitschr. f. Math. Logik u. Grundl. d. Math. **14** (1968) 357–376.

[20] J. Robinson, General recursive functions. *Proc. Am. Math. Soc.* **72** (1950) 703–718.

[21] H. Rogers Jr, *Theory of recursive functions and effective computability*. McGraw-Hill, New York (1967).

[22] H.E. Rose, *Subrecursion: functions and hierarchies*. Clarendon Press, Oxford (1984).

[23] J. Rothe, *Complexity theory and cryptology*. Springer-Verlag, Berlin (2005).

[24] H. Schwichtenberg, Rekursionszahlen und Grzegorczyk-Hierarchie. *Arch. Math. Logic* **12** (1969) 85–97.

[25] A.L. Selman, A survey of one-way functions in complexity theory. *Math. Syst. Theory* **25** (1992) 203–221.

[26] A. Selman, A taxonomy of complexity classes of functions. *J. Comput. Syst. Sci.* **48** (1994) 357–381.

[27] J.R. Shoenfield, On degrees of unsolvability. *Ann. Math.* **69** (1959) 644–653.

[28] R.I. Soare, *Recursively enumerable sets and degrees*. Springer-Verlag, Berlin (1987).

[29] R.I. Soare, *Computability and recursion. Bulletin of symbolic Logic* **2** (1996) 284–321.

[30] R.I. Soare, *Computability theory and applications*. Springer-Verlag, Berlin, forthcoming.

[31] K.W. Wagner, Bounded query classes. *SIAM Journal on Computing* **19** (1990) 833–846.

[32] G. Wechsung, *Vorlesungen zur Komplexitätstheorie*. B.G. Teubner, Stuttgart (2000).

[33] K. Weihrauch, *Computability*. Springer-Verlag, Berlin (1987).

Communicated by A. Bertoni.
Received August 4, 2009. Accepted August 20, 2010.