Filozofia Nauki (*The Philosophy of Science*)
ISSN 1230-6894
2018, vol. 26(4) [104]: 57-73
DOI: 10.14394/filnau.2018.0024

MICHAŁ WROCŁAWSKI*

REPRESENTING NUMBERS

Abstract

The purpose of this paper is to consider the question of how we can represent numbers (especially natural numbers) and how our choice of a representation affects our ability to compute various functions. In particular, we show the importance of computability of the characteristic function of identity in a representation of numbers. It turns out that it is a very strong assumption that significantly increases the scope of our knowledge about a given representation, including our ability to tell which functions are computable in this representation.

Keywords: representations of numbers, numerals, computable functions, characteristic functions, identity

When performing calculations on numbers, we need a way to represent them. A number can be represented by a set of objects of a certain cardinality (e.g., when we count on fingers), by words spoken aloud or in thoughts, diagrams (each bar represents a number proportional to its height). Even a single dot can represent a number in a certain situation — e.g., as a point on a complex plane. However, probably the most common way of representing numbers in mathematics is representing them as finite inscriptions (sequences of signs) and that is precisely the kind of representations that will concern us here.

We are unable to refer to numbers in any other way than by referring to a certain representation of them, either perceived by the senses or grasped in thought. To support this view, we can use the following argument: if there were no difference between a number and its representation, it would make no sense to say that the inscriptions "5," "V," and "five" all represent the same number.

* Institute of Philosophy, University of Warsaw, Krakowskie Przedmieście 3, 00-927 Warsaw, michalwro@wp.pl.

In this paper, we shall consider the question of computability of certain important functions, such as the successor function, addition, multiplication, and exponentiation, depending on the chosen representation of natural numbers. More specifically, one purpose of this paper is to show that, in every "reasonable" representation of natural numbers, we should be able to compute the characteristic function of identity: i.e., the function which decides whether two numerals represent the same number.

1. THE NOTION AND EXAMPLES OF REPRESENTATIONS

We shall begin by giving a definition of a representation of a set of numbers Z. Even though we are primarily concerned with representations of the set $\mathbb{N}$, our definition will allow that the represented set can be any $Z \subseteq \mathbb{C}$. We also want to allow the possibility that the same number can be represented by several (possibly infinitely many) numerals. We propose the following definition of a representation:

> DEFINITION 1. Let $Z \subseteq \mathbb{C}$ and A be a finite alphabet. We shall call (S, σ) a representation of numbers from the set Z, where $S \subseteq A^*$ is an infinite computable set, $Z \subseteq \mathbb{C}$ is countable, and $\sigma\colon S \to Z$ is a surjection.

For any representation (S, σ), the elements of A shall be called digits or symbols, and the elements of S shall be called numerals. We demand that Z should be countable because we are unable to represent uncountably many numbers with finite sequences over a finite (or even countable) alphabet.[1]

The simplest representation of the set $\mathbb{N}$ is the unary system, in which $A = \{\overline{1}\}$, S is the set of all finite sequences comprised of $\overline{1}$ and the empty word ε, and the function σ is defined as follows:

$$\sigma(\varepsilon) = 0,$$

$$\text{if } \sigma(\alpha^\frown\overline{1}) = n, \text{ then } \sigma(\alpha) = n + 1.$$

Throughout this paper, we adopt a convention according to which, for any natural number n, $\overline{n}$ refers to a numeral representing this number in the standard decimal representation.

[1] Let us notice that with a finite alphabet we can "imitate" an infinite set of symbols $\{a_i\colon i \in \mathbb{N}\}$. It suffices to take $A = \{a, \overline{1}\}$ and imitate every symbol a_i with a a numeral of the form $a\overline{1} \ldots \overline{1}$ with n occurrences of the digit $\overline{1}$.

DEFINITION 2. Let (S, σ) be a representation of the set Z. We shall say that this representation is unambiguous iff, for every $n \in Z$, there exists exactly one numeral $\alpha \in S$ such that $\sigma(\alpha) = n$. Otherwise, we shall call the representation ambiguous.

The unary representation of natural numbers defined above is unambiguous.

The starting point for our investigations here is Stewart Shapiro's paper "Acceptable Notation" (Shapiro 1982). Apart from irrelevant formal differences in definitions, Shapiro's notations can be equated with unambiguous representations of the set of natural numbers as defined in this paper. A part of the terminology and notations used throughout this paper comes from Konrad Zdanowski (2012), sometimes in a modified form.

Like Shapiro, we shall consider mostly representations of natural numbers. However, in contrast to his paper, we shall also include ambiguous representations. Such representations are very common in mathematics; for instance, both 2 + 6 and 8 represent the same number.

It might seem that the only "real" representation of a number is its standard decimal representation and all the terms, including functional predicates, are merely intermediate stages of a calculation leading to that "correct" representation of a number. We reject this position for the following reasons:

1. There is no reasons to assume that some letters, digits or symbols (or whatever we wish to call them), in this case functional predicates, should be arbitrarily rejected from the set of admissible symbols used to generate numerals. It seems that the status of all such symbols is exactly the same: they are signs on paper (or on another surface). The meaning of a sequence of symbols (a numeral) does not need to be the resultant of the meaning of individual symbols; it can be assigned to it by the function σ in a completely arbitrary way.

2. In many commonly used representations of numbers, the calculations performed on numbers represented by individual digits are somehow contained in numerals. For example, if a digit representing a smaller number stands after a digit representing a larger number in a Roman numeral, then we need to add them up. If their order is reversed, we need to subtract the smaller number from the larger number. According to this convention: VI = V + I = 6 and IV = V – I = 4. Another example is the convention of creating names for numerals in the German language. The inscription "zweiundvierzig" (literally: "two-and-forty") is the German name for the number 42. Thus, the most basic name of this number in German contains something we might call a functional predicate. There is no significant difference between + and "und." They both mean the same function.

3. In many cases, writing a number in its decimal representation need not be the optimal solution. The numeral $2{,}8 \cdot 10^{16}$ is much more readable than the numeral consisting of 28 followed by fifteen zeroes. In the same way, when we consider the divisibility of numbers, it is much more useful to know that "p is the hundredth prime number" than to know all the digits of its decimal representation. However, if, for whatever reason, we should need to calculate p^2, then we would prefer the latter representation. The conclusion is that there is no such thing as the "default" representation of a number; it all depends on the situation.

The matter is even more complicated when we go beyond natural numbers. There is no answer to the question of which representation is objectively better: $\sqrt{8}$ or $2\sqrt{2}$, $\frac{1}{3}$ or 0.(3), or $1\frac{1}{5}$ or $\frac{6}{5}$?

For the above reasons, we allow each number to be potentially represented by many (possibly infinitely many) numerals, none of which should be considered "standard." In other words, all such representations count as equally "valid."

Before we start a more formal part of this paper, let us take a brief look at yet another aspect of our definition of a representation: namely, the set of objects we wish to represent. We have decided that it is going to be an infinite computable subset of $\mathbb{C}$, but this is obviously not the only possibility. An important example worth mentioning is Church and Kleene's theory of notations of ordinal numbers (Church 1938, Kleene 1938; see also Rogers 1967). An interesting difference between Kleene's concept and ours is that Kleene does not assume that the set of all numerals must be computable. What he assumes is the computability of the $\chi_{\{0\}}$ function, which is the function that returns *TRUE* if and only if its argument is a numeral which represents 0, otherwise it returns *FALSE*.

2. COMPUTABILITY OF FUNCTIONS IN REPRESENTATIONS

In this section, we shall define some key notions necessary to describe properties of various representations and relations between representations.

REMARK 1. Since we have assumed that the alphabet is finite and the set of all numerals is computable, for each representation it is possible to:

1. decide whether it is a numeral of the considered representation for each sequence of symbols over the given alphabet,

2. generate an enumeration of all numerals of this representation (which, of course, does not have to be identical with the standard order on numbers). For the purposes of this paper, let us assume that, for any representation of numbers, we shall denote the consecutive numbers in this enumeration as: α_0, α_1, … .

> DEFINITION 3. Let (S, σ) be a representation of $Z \subseteq \mathbb{C}$. Then for any function F: $Z^r \rightarrow Z$, by F^σ: $S^r \rightarrow S$ we shall mean a function such that, for any $a_1, \ldots, a_r, b \in S$, the following condition is satisfied: $F^\sigma(a_1, \ldots, a_r) = b \Rightarrow \sigma(F(\sigma(a_1), \ldots, \sigma(a_r))) = \sigma(b)$.

Function F^σ shall be called an interpretation of the function F in the representation (S, σ). If there exists a computable function F^σ satisfying the above condition, then we shall say that F is computable in (S, σ).

REMARK 2. If the representation is ambiguous, then there can be many such functions. In other words, a function is computable in a given representation if there exists an algorithm which reads any numeral (or a finite sequence of numerals) on the input and returns any numeral which represents the value of the computed function on the output.

The above remark is closely related to the earlier assumption that every numeral representing a number is equally "valid" and that none of these numerals is privileged in any way.

The notion of computability of functions in representations defined above can sometimes turn out to be too weak. Suppose we have a representation (S, σ) of the set $\mathbb{N}$ in which the successor function is computable. Let us assume that an algorithm which computes this function reads a numeral α on the input and returns a numeral β on the output. Perhaps we wish to know if numeral β' (different from β) also represents the successor of the number represented by α. The assumption that the successor function is computable in this representation does not guarantee that we can find this out. This is why we shall introduce the notion of characteristic functions and define the computability of such functions in representations.

> DEFINITION 4. Let $R \subseteq Z^k$. The characteristic function of the relation R is the function χ_R such that for any $a_1, \ldots, a_k \in Z$ the following holds:
>
> $\chi_R(a_1, \ldots, a_k) = TRUE \Leftrightarrow R(a_1, \ldots, a_k)$,
>
> $\chi_R(a_1, \ldots, a_k) = FALSE \Leftrightarrow \sim R(a_1, \ldots, a_k)$.

In particular, we shall denote:

$$\chi(a_1, a_2) = TRUE \Leftrightarrow a_1 = a_2,$$

$$\chi(a_1, a_2) = FALSE \Leftrightarrow a_1 \neq a_2.$$

Let us adopt the following convention throughout this paper: whenever we speak of functions, unless explicitly stated otherwise, we mean only functions whose both arguments and values are numbers (in particular, they are not logical values).

> DEFINITION 5. Let (S, σ) be a representation of $Z \subseteq \mathbb{C}$. For any function f: $Z^r \to Z$, we shall say that the characteristic function χ_f is computable in (S, σ) iff there exists an algorithm which reads numerals $a_1, ..., a_r, \beta \in S$ on the input and returns *TRUE* or *FALSE* — the value of function $\chi_f(\sigma(a_1), ..., \sigma(a_r), (\beta))$ on the output.[2]

The above definition can be naturally extended to cover the computability of characteristic functions of any relations.

Let us consider the relation between the two notions of computability defined above.

THEOREM 1. Let (S, σ) be a representation of $Z \subseteq \mathbb{C}$ and f: $Z^k \to Z$, for a certain k $\in \mathbb{N}$. If χ_f is computable in (S, σ), then f is computable in it as well. If $\chi_=$ is also computable in this representation, then the implication in the opposite direction also holds.

PROOF. Let (S, σ) be a representation of Z and f be a function. Let us assume that χ_f is computable in this representation. We shall prove that f is also computable. Suppose we want to compute the value of $f^\sigma(a_1, ..., a_k)$ for some $a_1, ..., a_k \in S$.[3] We take a recursive enumeration of all numerals and we check them one by one until we find α_i such that $\chi_f^\sigma(a_1, ..., a_k, a_i) = TRUE$. Then $\alpha_i = f^\sigma(a_1, ..., a_k)$ is the value of the function we were looking for.

Now suppose that $\chi_=$ and f are computable in (S, σ). We shall prove that χ_f is computable as well. We want to find the value of $\chi_f^\sigma(a_1, ..., a_k, b)$ for certain $a_1, ..., a_k \in S$. Since f is computable, we can find a numeral c such that $\sigma(f(a_1, ..., a_k)) = \sigma(c)$. Then:

$$\chi_f(a_1, ..., a_k, b) = TRUE \Leftrightarrow \sigma(b) = \sigma(c),$$

$$\chi_f(a_1, ..., a_k, b) = FALSE \Leftrightarrow \sigma(b) \neq \sigma(c). \blacksquare$$

[2] It is important to emphasize that *TRUE* and *FALSE* are not symbols from the alphabet A, nor from the set S, but they are additional symbols representing logical values.

[3] If (S, σ) is ambiguous, then obviously it is possible that there are many such functions. However, we just need to find the value for any of these functions.

3. CRITERIA FOR SIMILARITY OF REPRESENTATIONS

We want to discuss the question: when can two representations be considered similar? In mathematics, two objects that are "nearly the same" (they have the same structure) are commonly referred to as isomorphic. Mathematical investigations are often carried out "up to isomorphism;" thus, isomorphic objects are equated with each other.

We shall say that two representations are isomorphic if there exist computable functions F_τ^σ and F_σ^τ which shall be called translations between these representations.[4]

DEFINITION 6. Representations (S, σ) and (T, τ) of $Z \subseteq \mathbb{C}$ are isomorphic iff there exist computable functions F_τ^σ and F_σ^τ such that:

$$\forall_{a \in S} \exists_{b \in T} (F_\tau^\sigma(a) = b \wedge \sigma(a) = \tau(b)),$$

$$\forall_{b \in T} \exists_{a \in S} (F_\sigma^\tau(a) = b \wedge \sigma(a) = \tau(b)).$$

However, this definition raises some problems. Let (S, σ) and (T, τ) be representations of a certain set Z. Let $S = \{\alpha_i : i \in \mathbb{N}\}$ and $T = \{\beta_i : i \in \mathbb{N}\}$. Suppose that algorithms computing translations F_τ^σ and F_σ^τ, respectively, read α_i on the input and return β_i on the output, or read β_i on the input and return α_i on the output (for any natural number i). For example, an algorithm that calculates F_τ^σ reads numeral α_6 on the input and returns β_6. Suppose, however, that numerals β_6 and β_8 represent the same number in (T, τ). The assumption that both representations are isomorphic does not guarantee that it will be possible to establish that that is the case. For this reason, we define another criterion.

DEFINITION 7. Representations (S, σ) and (T, τ) are strongly isomorphic iff there exists a computable function $Id_\tau^\sigma : S \times T \rightarrow \{TRUE, FALSE\}$ such that for any $a \in S$, $b \in T$:

$$Id_\tau^\sigma(a, b) = TRUE \Leftrightarrow \sigma(a) = \tau(b),$$

$$Id_\tau^\sigma(a, b) = FALSE \Leftrightarrow \sigma(a) \neq \tau(b).$$

Of course, we would like to know what the relation is between isomorphism and strong isomorphism of representations.

[4] It is important that we demand that these functions should be computable. Without this assumption, any two representations would be isomorphic, thus making this notion utterly useless.

THEOREM 2. If representations (S, σ) and (T, τ) are strongly isomorphic, then they are isomorphic.

PROOF. Suppose that representations (S, σ) and (T, τ) are strongly isomorphic. We want to show that they are isomorphic. We shall show that there exists a computable translation F^{σ}_{τ} from (S, σ) to (T, τ) (the proof of existence of a translation in the opposite direction is analogous).

Let $a \in S$. Since T is computable, and thus recursively enumerable, we check all the numerals from this set one by one until we find such $b \in T$ that $Id^{\sigma}_{\tau}(a, b) = TRUE$ (we can check it because we assumed that these representations are strongly isomorphic). Then the translation returns b on the output. ■

It turns out that the implication in the opposite direction does not hold.

THEOREM 3. There exist representations (S, σ) and (T, τ) that are isomorphic, but not strongly isomorphic.

Examples of such representations are direct consequences of Theorem 5. We will show now that — just like in the case of two notions of computability — both notions of isomorphism also become equivalent if we assume that the characteristic function of identity is computable.

THEOREM 4. For any representations (S, σ) and (T, τ) of $Z \subseteq \mathbb{C}$ with computable function $\chi_=$, these representations are strongly isomorphic iff they are isomorphic.

PROOF. The implication ($\Rightarrow$) has already been proved for all representations. We shall now prove the implication ($\Leftarrow$). Suppose that (S, σ) and (T, τ) are isomorphic. Let $F^{\sigma}_{\tau}(a, b)$ be a computable translation from (S, σ) to (T, τ). Let $a \in S$ and $b \in T$. We would like to know if $\sigma(a) = \tau(b)$. We use an algorithm (which we have assumed to exist) to find $b' = F^{\sigma}_{\tau}(a)$. Since $\chi_=$ is computable in (T, τ), we can calculate $Id^{\sigma}_{\tau}(a, b)$ in the following way: $Id^{\sigma}_{\tau}(a, b) = \chi_=(b, b')$. ■

REMARK 3. In Theorem 4, it suffices to assume that $\chi_=$ is computable in any of the representations under consideration because — as we shall prove later — in isomorphic representations, exactly the same functions are computable.

We shall now prove several properties of the relations between representations defined above.

THEOREM 5. If $\chi_=$ is not computable in (S,σ), then (S,σ) is not strongly isomorphic to any representation.

PROOF. Let (S, σ) be a representation in which $\chi_=$ is not computable. Let (T, τ) be any representation. Suppose that these two representations are strongly isomorphic. Then the following functions are computable: Id^{σ}_{τ} and translations between these representations F^{σ}_{τ} and F^{τ}_{σ}. For any given a, $a' \in S$, we want to find the value of $\chi_= (a, a')$. However, we assumed in particular that we can calculate $F^{\sigma}_{\tau}(a')$ and $Id^{\sigma}_{\tau}(a, F^{\sigma}_{\tau}(a'))$.

The following are equivalent:

$$Id^{\sigma}_{\tau}(a, F^{\sigma}_{\tau}(a')) = TRUE,$$

$$\sigma(a) = \tau(F^{\sigma}_{\tau}(a')),$$

$$\sigma(a) = \sigma(a'),$$

$$\chi_=(a, a') = TRUE.$$

REMARK 4. In particular, it follows from Theorem 5 that such a representation is not strongly isomorphic even to itself.

DEFINITION 8. Representations (S, σ) and (T, τ) are equivalent iff for every function f the following condition holds: f is computable in $(S, \sigma) \Leftrightarrow f$ is computable in (T, τ).

DEFINITION 9. Representations (S, σ) and (T, τ) are χ-equivalent iff for every function f the following condition holds: χ_f is computable in $(S, \sigma) \Leftrightarrow \chi_f$ is computable in (T, τ).

THEOREM 6. Let $Z \subseteq \mathbb{C}$ and let (S, σ) and (T, τ) be representations of Z. If these representations are isomorphic, then they are equivalent.

PROOF. Let $Z \subseteq \mathbb{C}$ and let (S, σ) and (T, τ) be representations of Z. Let F^{σ}_{τ} be a translation from (S, σ) to (T, τ) and F^{τ}_{σ} — a translation from (T, τ) to (S, σ). Let f: $Z \rightarrow Z$ be computable in (S, σ) (without loss of generality we assume that f is unary). We shall prove that f is computable in (T, τ).

Let $b \in T$. We calculate $f(\tau(b))$ in (T, τ). We know that there exists $a \in S$ such that $F^{\tau}_{\sigma}(b) = a$ and it follows from our assumption that we can find such a. Let $a' = f^{\sigma}(a)$, where f^{σ} is a computable function which represents f in (S, σ), and let $b' = F^{\sigma}_{\tau}(a')$ (both these functions are computable due to our assumptions). Let us denote $n = \sigma(a) = \tau(b)$ and $n = \sigma(a') = \tau(b')$. Then $n' = f(n)$. It follows that $f(\tau(b)) = n'$. But there is an algorithm which reads $b \in T$ on the input and returns $b' \in T$ such that $f(\tau(b)) = \tau(b'))$. Therefore, f is computable in (T, τ).

The proof of the implication in the opposite direction is analogous. ■

THEOREM 7. Let $Z \subseteq \mathbb{C}$ and let (S, σ) and (T, τ) be representations of Z. If these representations are isomorphic, then they are χ-equivalent.

PROOF. Let representations (S, σ) and (T, τ) be isomorphic and let function f be such that χ_f is computable in (S, σ). We shall prove that f is also computable in (T, τ).

Since both representations are isomorphic, there exists a computable translation F_σ^τ: $T \Rightarrow S$. Then for any a, $b \in T$ the following holds: $\chi_f^\tau(a, b) = \chi_f^\sigma(F_\sigma^\tau(a), F_\sigma^\tau(b))$, where χ_f^σ and χ_f^τ are functions interpreting χ_f respectively in representations (S, σ) and (T, τ) (without loss of generality we can assume that f is unary). Since χ_f^σ and F_σ^τ are computable, we conclude that χ_f^τ is also computable. ■

THEOREM 8. For any representation (S, σ) of Z with computable function $\chi_=$ there exists an unambiguous representation (T, τ) of Z strongly isomorphic to (S, σ).

PROOF. Let (S, σ) be a representation. Let $\alpha_0, \alpha_1, \alpha_2, \ldots$ be a computable enumeration of all numerals from S. We define an infinite sequence of sets $T_0, T_1, T_2, \ldots$ as follows:

$$T_0 = \varnothing,$$

$$T_{n+1} = \begin{cases} T_n \text{ if } \exists_{i \le n} \sigma(\alpha_i) = \sigma(\alpha_n), \\ \\ T_n \cup \{\alpha_{n+1}\}, \text{otherwise.} \end{cases}$$

Let $T = \bigcup_{n \in \mathbb{N}} T_n$ and $\tau = \sigma \cap (T \times Z)$. We shall prove that (T, τ) is an unambiguous representation of Z strongly isomorphic to (S, σ).

First, we shall prove that (T, τ) is a representation of Z. To this end, we shall prove that all the conditions from the definition of a representation hold.

> *Condition 1*: $T \subseteq A^*$, where A is finite. That is the case because $T \subseteq S \subseteq A^*$, where A is finite.
>
> *Condition 2*: T is computable. Let $\alpha \in A^*$. We want to check if $\alpha \in T$. We can check if $\alpha \in S$.

If $\alpha \notin S$, then $\alpha \notin T$ because $T \subseteq S$. If $a \in S$, then there exists $n \in \mathbb{N}$ such that $\alpha = \alpha_n$. Let us check numerals in the sequence $\{\alpha_n\}_{n \in \mathbb{N}}$ one by one until we find such n. In which case: $\alpha_n \in T \Leftrightarrow \exists_{i \le n}\, \sigma(\alpha_i) = \sigma(\alpha_n)$.

We can find out if this condition is satisfied because we only need to check if $\sigma(\alpha_i) = \sigma(\alpha_n)$ holds for finitely many numerals. This is computable because we assumed that $\chi_=$ is computable in (S, σ). Thus, T is computable.

Condition 3: $\tau : T \Rightarrow Z$ is a function onto Z.

It is obvious that $\tau = \sigma \cap (T \times Z)$ is a function. We want to show that this function is onto Z.

Let $a \in Z$. Since σ is onto Z, we conclude that, for a certain natural number n, the following holds: $\sigma(\alpha_n) = a$. If $\alpha_n \in T$, then $\tau(\alpha_n) = a$. Otherwise, there exists $i \leq n$ such that $\sigma(\alpha_i) = \sigma(\alpha_n)$. Let us take the smallest such i. It follows from the definition that $\alpha_i \in T_i \subseteq T$. Therefore, we conclude that $\alpha_i \in T$ and $\tau(\alpha_i) = a$. It follows that τ is onto Z. Therefore, (T, τ) is a representation of Z.

It is a simple conclusion from the definition of the sequence T_0, T_1, T_2, ... that this representation must be unambiguous. It is also strongly isomorphic to (S, σ). For, let us take any $a \in S$ and $b \in T \subseteq S$. Since $T \subseteq S$, it follows that: $Id_\tau^\sigma(a, b) = \chi_=(a, b)$. However, we assumed that $\chi_=$ is computable in (S, σ). ■

What conclusions can be drawn from the above theorems? Undoubtedly, strong isomorphism is the strongest of the criteria of similarity of representations we have considered. It seems that we can claim that if two representations are strongly isomorphic, then — from our point of view — they are "nearly identical."[5]

However, this "near identity" is considered from a computational or — philosophically speaking — cognitive perspective. It assumes that two representations are not only similar (have a similar structure), but also that we can determine (compute) what this similarity consists in — i.e., which numerals in one representation correspond to which numerals in the other. It turns out that such a strong notion is only relevant when we consider representations with the computable characteristic function of identity. As a summary of this part of the paper, let us formulate the following conclusions:

1. Strong isomorphism is the strongest of all the criteria of similarity of representations formulated here.

2. If, in a given representation, the characteristic function of identity is not computable, then the representation is not strongly isomorphic to any representation. In other words: we have no strong criterion to compare it with other representations.

3. If, however, in a given representation, the characteristic function of identity is computable, then the representation is strongly isomorphic to a certain

[5] Of course, such "near identity" is a relative notion. It depends on what properties of representations we want to preserve. E.g., from the point of view of computational complexity, two strongly isomorphic representations can differ significantly. However, we are not concerned with such issues in this paper.

unambiguous representation. If we assume that a strong isomorphism preserves all the important properties of representations, then it follows that, as long as we consider only representations where $\chi_=$ is computable, allowing ambiguous representations is insignificant in the following sense: such representations do not possess any important properties other than those already possessed by some unambiguous representations. Although departing from the domain of representations with computable function $\chi_=$ can significantly broaden the scope of our inquiry, our ability to compare such representations with each other is unfortunately very limited.

4. COMPUTABILITY OF FUNCTIONS ON NATURAL NUMBERS IN VARIOUS REPRESENTATIONS

We want to consider the question of the computability of various functions in a given representation if we know that certain other functions are computable in it. In this part of the paper, we limit ourselves to natural numbers. While research in this area is still under way, here we give some examples of when the assumption of the computability of $\chi_=$ makes a difference when considering the computability of certain other functions.

THEOREM 9. Let (S, σ) be a representation of $\mathbb{N}$. The following conditions are equivalent:

(1) (S, σ) is isomorphic to the standard representation of $\mathbb{N}$.

(2) (S, σ) is strongly isomorphic to the standard representation of $\mathbb{N}$.

(3) The successor function and $\chi_=$ are computable in (S, σ).

PROOF. The equivalence of (1) and (2) follows from Theorem 4 because $\chi_=$ is computable in the standard representation of $\mathbb{N}$.

The implication (1) $\Rightarrow$ (3) follows from Theorems 6 and 7 because the successor function and $\chi_=$ are computable in the standard representation of $\mathbb{N}$ and computability of all functions (both numerical and characteristic) is preserved by isomorphism of representations.

We shall now prove the implication (3) $\Rightarrow$ (1). Let (S, σ) be a representation of $\mathbb{N}$ in which the successor function (denoted as *Succ*) and $\chi_=$ are computable. In this representation, there is a numeral representing number 0. Let us denote such a numeral by α.

We shall show how to compute translations between (S, σ) and the standard representation. Let λ be a numeral of (S, σ). For every natural number i, let us denote $\lambda_i = Succ^{\sigma}(Succ^{\sigma}(\dots(\alpha)\dots))$, where the successor is iterated i times in λ_i. We compare one by one each λ_i with λ until we find i such that $\sigma(\lambda) = \sigma(\lambda_i)$. In which case $\sigma(\lambda) = i$, so the numeral $\overline{i}$ represents the same number in the standard representation as the numeral λ in (S, σ).

Let $\overline{n}$ be any numeral of the standard representation (denoting — according to the convention — number n). To find its counterpart in (S, σ), we calculate λ_n defined as above. ■

CONCLUSION. Let (S, σ) be a representation of $\mathbb{N}$ in which the successor function and $\chi_=$ are computable. Then, in such a representation, exactly those functions are computable which are computable in the standard representation of $\mathbb{N}$, including, in particular, addition, multiplication, and exponentiation.

PROOF. It is an immediate conclusion from Theorems 6 and 9. ■

THEOREM 10. There exists a representation (S, σ) of $\mathbb{N}$ in which the successor function is computable, but addition, multiplication, and exponentiation are not computable.

PROOF. We construct (S, σ) as follows:
The alphabet consists of symbols: $\overline{0}$, $\overline{1}$, a. The numerals are all finite non-empty sequences of symbols from the alphabet which contain at most one occurrence of a.

Let $Z \subseteq \mathbb{N}$ be uncomputable in the standard representation. We construct σ in the following way:

$$\sigma(\overline{0}) = 0,$$

$$\sigma(\overline{1}) = 1,$$

$$\sigma(a) = 0 \Leftrightarrow 1 \notin Z,$$

$$\sigma(a) = 1 \Leftrightarrow 1 \in Z.$$

Also, for any $\alpha \in S$:

$$\sigma(\alpha^\frown\overline{0}) = \sigma(\alpha),$$

$$\sigma(\alpha^\frown\overline{1}) = \sigma(\alpha) + 1,$$

$$\sigma(\alpha^\frown a) = \sigma(\alpha) \Leftrightarrow lh(\alpha) = n \wedge n + 1 \notin Z,$$

$$\sigma(\alpha^\frown a) = \sigma(\alpha) + 1 \Leftrightarrow lh(\alpha) = n \wedge n + 1 \in Z,$$

where $^\frown$ is a concatenation and $lh\,(\alpha)$ is the length of the sequence α.

This is a correct representation because every natural number n is represented by at least one numeral, namely $\overline{1} \dots \overline{1}$ consisting of n digits $\overline{1}$ with the exception of number 0, which is represented by the numeral $\overline{0}$. The successor function in (S, σ) is defined as follows: $\mathrm{Succ}(\alpha) = \alpha^\frown\overline{1}$. This function is computable.

We shall show that addition is not computable in this representation. In this part of the proof, for any natural number $n \geq 1$, let us denote: $\lambda_n = \overline{0} \dots \overline{0}a$, where λ_n consists of $n - 1$ digits $\overline{0}$ followed by one occurrence of a. For any sequence $\alpha \in S$ and any symbol $b \in A$, let $\#_b(\alpha)$ denote the number of occurrences of b in the numeral α. We want to find out whether $n \in Z$.

We compute $\lambda_n + \lambda_n$ in (S, σ). We know that $\sigma(\lambda_n)$ is equal to 0 or 1. Thus, $\sigma(\lambda_n +^\sigma \lambda_n)$ is equal to 0 or 2. If $n \in Z$, then $\sigma(\lambda_n) = 1$ and $\sigma(\lambda_n +^\sigma \lambda_n) = 2$. Then, there are the following possibilities:

$$\#_{\overline{1}}(\lambda_n +^\sigma \lambda_n) = 1 \wedge \#_a(\lambda_n +^\sigma \lambda_n) = 1$$

or

$$\#_{\overline{1}}(\lambda_n +^\sigma \lambda_n) = 2 \wedge \#_a(\lambda_n +^\sigma \lambda_n) = 0.$$

If, however, $n \notin Z$, then $\sigma(\lambda_n) = \sigma(\lambda_n +^\sigma \lambda_n) = 0$ and then $\#_{\overline{1}}(\lambda_n +^\sigma \lambda_n) = 0$.

It is easy to find out which of these is the case and thus — whether $n \in Z$. It follows that Z is computable in the standard representation, which contradicts our assumption. Therefore, addition is not computable in (S, σ).

Similarly, we show that multiplication and exponentiation are not computable in (S, σ). Let us denote: $\lambda_n = \overline{1} \dots \overline{1}a$, where λ_n consists of $n - 1$ digits $\overline{1}$ followed by one occurrence of a. Then we compute, respectively, $\lambda_n \cdot \lambda_n$ or $\lambda_n^{\overline{11}}$ in (S, σ) (notice that they both return the same result; we shall only provide a proof for multiplication).

Assume that multiplication is computable in (S, σ). We shall prove that Z is also computable then. Let $n \in \mathbb{N}$. We want to find out whether $n \in Z$. Without loss of generality, we can assume that $n \geq 2$.[6] Let $\alpha \in S$ be the result of multiplication $\lambda_n \cdot \lambda_n$ in (S, σ). We know that $\sigma(\lambda_n)$ is equal to either $n - 1$ or n. Therefore:

If $\sigma(\lambda_n) = n - 1$, then $\sigma(\lambda_n \cdot^\sigma \lambda_n) = (n-1)^2 = n^2 - 2n + 1$. Therefore, $\#_{\overline{1}}(\alpha) = n^2 - 2n$ or $\#_{\overline{1}}(\alpha) = n^2 - 2n + 1$.

[6] The algorithm which is supposed to find out whether $n \in Z$ will have answers for $n \in \{0, 1\}$ explicitly given as special cases.

If $\sigma(\lambda_n) = n$, then $\sigma(\lambda_n \cdot^{\sigma} \lambda_n) = n^2$. Therefore, $\#_{\bar{1}}(\alpha) = n^2 - 1$ or $\#_{\bar{1}}(\alpha) = n^2$.

Notice that, for $n \geq 2$, we can find out which of these cases occurs. If the former is the case, then $n \notin Z$; otherwise $n \in Z$. Thus, we have obtained contradiction with the assumption that Z is not computable. Therefore, multiplication (and similarly exponentiation) is not computable in (S, σ). ∎

THEOREM 11. For any $C \subseteq \mathbb{N}$, there exists a function f on natural numbers such that, for every representation (S, σ) of $\mathbb{N}$ with both f and $\chi_=$ computable, χ_C is also computable in (S, σ).

PROOF. Let $C \subseteq \mathbb{N}$. We shall construct f: $\mathbb{N} \to \mathbb{N}$ in the following way:

$$f(n) = \begin{cases} 1 \text{ } if \text{ } n \in C, \\ \\ 0 \text{ } if \text{ } n \notin C. \end{cases}$$

Suppose that f and $\chi_=$ are both computable in (S, σ). Let $\lambda \in S$. We shall show how to find out if $\sigma(\lambda) \in C$. Let α be a certain numeral representing 1 in (S, σ). Suppose that an algorithm calculating f in (S, σ) read λ on the input and returned β on the output. We need to establish whether β is a numeral representing 0 or 1. The following are equivalent:

$\chi_=(\alpha, \beta) = TRUE$,

$\sigma(\beta) = 1$,

$f(\sigma(\lambda)) = 1$,

$n \in C$,

$\chi_C = TRUE$.

Thus, if $\chi_=(\alpha, \beta) = TRUE$, then $\chi_C = TRUE$. Similarly, if $\chi_=(\alpha, \beta) = FALSE$, then $\chi_C = FALSE$. We conclude that $\chi_C = \chi_=(\alpha, \beta) = \chi_=(f^{\sigma}(\lambda), \beta)$. It follows that χ_C is computable in (S, σ). ∎

THEOREM 12. For any at most countable set of functions on natural numbers F and any $C \subseteq \mathbb{N}$ such that $C \neq \varnothing$, $C \neq \mathbb{N}$, there exists a representation (S, σ) of $\mathbb{N}$ in which all functions from F are computable, but χ_C is not computable.

PROOF. Without loss of generality, let us assume that all functions in F are unary and that $F = \{f_i\}_{i \in \mathbb{N}}$ is an infinite countable set. We construct (S, σ) as follows: The alphabet A consists of standard digits $\overline{0}$, ..., $\overline{9}$, symbol $\overline{f}$, symbols (,) and the comma.

All numerals of the standard representation are also numerals of (S, σ). For any $\lambda \in S$ and any natural number i: $\overline{f}\,\overline{1} \ldots \overline{1}(\lambda) \in$ S, where the number of occurrences of symbol $\overline{1}$ equals i.

Let $Z \subseteq \mathbb{N}$ be any set uncomputable in the standard representation. Both Z and $\mathbb{N} - Z$ must be infinite then. Let $a_0, a_1, \ldots$ be an enumeration of all numbers from Z in ascending order and $b_0, b_1, \ldots$ — of all numerals from $\mathbb{N} - Z$.

We define σ on standard numerals as follows: each numeral $\overline{a_i}$ represents a certain number from C, each numeral from $\overline{b_i}$ represents a certain number from $\mathbb{N} - C$, and every natural number is represented by at least one numeral $\overline{a_i}$ or $\overline{b_i}$. Such an assignment exists because Z and $\mathbb{N} - Z$ are both infinite while C and $\mathbb{N} - C$ are both non-empty. Also, for any natural number i and any $\lambda \in S$, let: $\sigma(\overline{f}\,\overline{1} \ldots \overline{1}(\lambda)) = f_i(\sigma(\lambda))$ where there are exactly i occurrences of symbol $\overline{1}$ in $\overline{f}\,\overline{1} \ldots \overline{1}$.

This representation is well-defined because each natural number is represented by at least one numeral; namely, a certain numeral of the standard representation. For any function $f_i \in F$ and any $\lambda \in S$, we define: $(f_i)^{\sigma}(\lambda) = \overline{f}\,\overline{1} \ldots \overline{1}(\lambda)$, where there are exactly i occurrences of symbol $\overline{1}$ in $\overline{f}\,\overline{1} \ldots \overline{1}$. It follows that all the functions from F are computable in (S, σ).

We shall show that χ_C is not computable in (S, σ). Suppose, to the contrary, that it is computable. We shall show that this assumption leads to a contradiction because then the set Z would also have to be computable.

It follows from the assumptions that, for any natural number n: $n \in Z$ iff $\sigma(\overline{n}) \in C$, so $n \in Z$ iff $\chi_C(\overline{n}) = TRUE$. According to our assumption, χ_C is computable. Thus, Z is also computable, which leads to a contradiction. Therefore χ_C is not computable. ∎

SUMMARY

The purpose of this paper was to argue in favour of the following claim: when choosing a representation of numbers, it is very important that the characteristic function of identity $\chi_=$ should be computable in it. Here is a summary of the arguments supporting this claim:

1. If we assume the computability of $\chi_=$, certain weaker notions become equivalent to certain stronger notions: the computability of f and of χ_f (Theorem 1) as well as isomorphism and strong isomorphism (Theorem 3). It is

therefore possible to simplify the conceptual apparatus used to describe and compare representations.

2. Representations without computable function $\chi_=$ cannot be strongly isomorphic to any representations (Theorem 5). Our ability to compare them with each other is thus limited, as is our ability to compare numerals from various such representations.

3. The computability of $\chi_=$ together with the computability of a certain given function in a representation of $\mathbb{N}$ can, in some cases, be a sufficient condition of computability of certain other functions, while if we do not assume the computability of $\chi_=$, the computability of those other functions does not follow from our assumptions (Theorems 10 and 12).

BIBLIOGRAPHY

Church A. (1938), "The Constructive Second Number Class," *Bulletin of the American Mathematical Society* 44, 224-232.

Kleene S. (1938), "On Notation for Ordinal Numbers," *The Journal of Symbolic Logic* 3, 150-155.

Rogers H. Jr. (1967), *Theory of Recursive Functions and Effective Computability*, Cambridge, MA: MIT.

Shapiro S. (1982), "Acceptable Notation," *Notre Dame Journal of Formal Logic* 23(1), 14-20.

Zdanowski K. (2012), *On Notation Systems for Natural Numbers and Polynomial Time Computations*, Numbers and Truth, Gothenburg (unpublished slides from a conference).