# Computability of functions and sets in various representations of natural numbers

Michał Wrocławski

Institute of Philosophy, University of Warsaw

*michalwro@wp.pl*

24 July 2018

# Overview

# Introduction

1. When performing calculations on numbers, we need to represent them somehow.
2. We usually represent numbers as some inscriptions (finite sequence of symbols referred to as digits).
3. Here we are going to be primarily concerned with representing natural numbers.
4. Among most common representation of natural numbers are: the decimal representation, the binary representation and the representation by Roman numerals.

# Definition of a representation

**Definition**

Let $A$ - a finite alphabet. We shall call $(S, \sigma)$ a representation of numbers from $\mathbb{N}$, where $S \subseteq A^*$ is an infinite computable set and $\sigma : S \to \mathbb{N}$ is a surjection.

**Remark**

Notice that according to this definition, a number can be represented by many (possibly infinitely many) numerals.

**Definition**

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

# Examples

## Decimal representation (standard representation)

Let $A = \{\bar{0}, \bar{1}, ..., \bar{9}\}$. $S$ shall be the set consisting of numeral $\bar{0}$ and of all finite inscriptions consisting of symbols from $A$ which do not start with $\bar{0}$. To each numeral of the form $\overline{a_i}...\overline{a_0}$, the function $\sigma$ assigns the number $\sum_{i=0}^{n} a_i \cdot 10^i$.

## Binary representation

Let $A = \{\bar{0}, \bar{1}\}$. $S$ shall be the set consisting of numeral $\bar{0}$ and of all finite inscriptions consisting of symbols from $A$ which do not start with $\bar{0}$. To each numeral of the form $\overline{a_i}...\overline{a_0}$, the function $\sigma$ assigns the number $\sum_{i=0}^{n} a_i \cdot 2^i$.

# Computability of functions in representations

1. We are going to distinguish between two types of functions: numerical functions and characteristic (Boolean functions).
2. Numerical functions take numbers (or finite sequences of numbers) as arguments and assign other numbers to them.
3. Characteristic functions take numbers (or finite sequences of numbers) as arguments and assign to them logical values *TRUE* and *FALSE* which we view as entities entirely separate from both numbers and numerals.

# Computability of numerical functions in representations

## Definition of computability of numerical functions

Let $(S, \sigma)$ be a representation of $\mathbb{N}$. Then for any function $F : \mathbb{N}^r \to \mathbb{N}$, by $F^\sigma : S^r \to S$ we shall mean a function such that for any $a_1, ..., a_r, b \in S$ the following condition is satisfied:

$$F^\sigma(a_1, ..., a_r) = b \Rightarrow \sigma(F(\sigma(a_1), ..., \sigma(a_r))) = \sigma(b).$$

The function $F^\sigma$ shall be called an interpretation of the function $F$ in the representation $(S, \sigma)$. If there exists a computable function $F^\sigma$ satisfying the above condition, than we shall say that $F$ is computable in $(S, \sigma)$.

# Computability of characteristic functions in representations

## Definition of computability of characteristic functions

Let $R \subseteq \mathbb{N}^k$. The characteristic function of the relation $R$ is the function $\chi_R$ such that for any $a_1, ..., a_k \in \mathbb{N}$ the following holds:

$$\chi_R(a_1, ..., a_k) = TRUE \Leftrightarrow R(a_1, ...a_k).$$

$$\chi_R(a_1, ..., a_k) = FALSE \Leftrightarrow \neg R(a_1, ...a_k).$$

In particular, we shall denote:

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

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

# 3 types of representations

We can divide representations of natural numbers into 3 categories:

1. Representations where each number is represented by exactly one numeral (unambiguous representations)
2. Representations where some numbers are represented by more than one numeral, but $\chi_=$ is computable.
3. Representations where $\chi_=$ is not computable.

It appeared that:

1. Each representation of the second type is equivalent to a certain representation of the first type in a certain very strong sense which I call strong isomorphism.
2. Representations of the third type can have some very strange properties so it is very hard to formulate any general theorems about them.

# Computability of functions in various representations - examples

**Theorem**

Let $(S, \sigma)$ 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 $\mathbb{N}$, in particular addition, multiplication and exponentiation.

**Theorem**

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

# Computability of functions in various representations - examples

## Proof (sketch)

We construct $(S, \sigma)$ 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 $\sigma$ in the following way:

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

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

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

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

# Computability of functions in various representations - examples

## Proof (sketch) - continued

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 $\alpha$.

# Computability of numerical vs characteristic functions

There are more similar theorems where the assumption of computability of $\chi_=$ influences computability of certain important numerical functions.
However, surprisingly, no assumptions regarding computability of numerical functions can guarantee us computability of nontrivial characteristic functions.
Here I formulate a theorem and sketch the proof. However, this theorem can be generalised to cover even more cases.

# Computability of numerical vs characteristic functions

**Theorem**

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

## Proof (sketch)

We assume that all functions are unary and that $(F_i)_{i \in \mathbb{N}}$ is an infinite countable family.

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 $(S, \sigma)$.

For any $\lambda \in S$ and any function $f_i \in F$: $\overline{f} \underbrace{\overline{1}...\overline{1}}_{i \text{ times}} (\lambda) \in S$.

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

We define $\sigma$ on standard numerals as follows: each numeral $\overline{a_i}$ represents a certain number from $C$ and each numeral from $\overline{b_i}$ - a certain number from $\mathbb{N} - C$, and every natural number is represented by at least one numeral $\overline{a_i}$ or $\overline{b_i}$.

## Proof (sketch) - continued

Also, for any natural number $i$ and any $\lambda \in S$, let:

$$\sigma(\overline{f}\ \underbrace{\overline{1}...\overline{1}}_{i\text{ times}}(\lambda)) = f_i(\sigma(\lambda)).$$

We shall show that $\chi_C$ is not computable in $(S, \sigma)$. Suppose to the contrary that it is 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 $\chi_C$ is computable. Thus, $Z$ is also computable, which leads to a contradiction. Therefore $\chi_C$ is not computable.

1. The choice of a representation is crucial when performing computations.
2. By choosing a certain representation, we can manipulate which functions are going to be computable in it and which are not.
3. The key role of computability of $\chi_=$ for representations of natural numbers - we can say much more about representations in which this function is computable. Without this assumption the computability of different functions is to a large extent independent from each other.

# References


Stewart Shapiro (1982)
Acceptable notation
*Notre Dame Journal of Formal Logic* 23(1):14–20, 01 1982.

Michael Rescorla (2007)
Church's Thesis and the conceptual analysis of computability
*Notre Dame Journal of Formal Logic* 48:253-280, 2007.

Jack Copeland and Diane Proudfoot (2010)
Deviant encodings and Turing's analysis of computability
*Studies in History and Philosophy of Sciences* 41:2473-252, 2010.

# The End