



**Research article**

## Numerical computations for Operator axioms

**Pith Peishu Xie\***

Axiom Studio, PO Box #3620, Jiangdongmen Postoffice, Gulou District, Nanjing 210036, China

\* **Correspondence:** Email: pith.xie@outlook.com.

**Abstract:** The Operator axioms have produced new real numbers with new operators. New operators naturally produce new equations and thus extend the traditional mathematical models which are selected to describe various scientific rules. So new operators help to describe complex scientific rules which are difficult described by traditional equations and have an enormous application potential. As to the equations including new operators, engineering computation often need the approximate solutions reflecting an intuitive order relation and equivalence relation. However, the order relation and equivalence relation of real numbers are not as intuitive as those of base-b expansions. Thus, this paper introduces numerical computations to approximate all real numbers with base-b expansions.

**Keywords:** numerical computation; numerical analysis; Operator axiom

**Mathematics Subject Classification:** 11Y16, 65H05, 49M25, 11B85, 03D05

---

### 1. Introduction

In [1], we distinguished the limit from the infinite sequence. In [2], we defined the Operator axioms to extend the traditional real number system. In [3], we promoted the research in the following areas:

1. We improved on the Operator axioms.
2. We defined the VE function and EV function. For clarity, we rename VE Function [3] to Prefix Function. For clarity, we rename EV Function [3] to Suffix Function.
3. We proved two theorems about the Prefix function and Suffix function.

The Operator axioms forms a new arithmetic axiom. The [3, Definition 2.2] defines real number system on the basis of the logical calculus  $\{\Phi, \Psi\}$ . The [2, TABLE 2] defines new operators according to the definition of number systems. Real operators naturally produce new equations such as  $y = [x + + + + [1 + 1]]$ ,  $y = [[1 + 1] - - - - x]$ ,  $y = [x // // [1 + 1]]$  and so on. In other words, real operators extend the traditional mathematical models which are selected to describe various scientific rules. Operator axioms have included infinite operators, so no other operator can be added to them. This means the Operator axioms is a complete real number system. In fact, infinite operators imply the

completeness.

Thus, real operators exhibit potential value as follows:

1. Real operators can give new equations and inequalities so as to precisely describe the relation of mathematical objects.

2. Real operators can give new equations and inequalities so as to precisely describe the relation of scientific objects.

So real operators help to describe complex scientific rules which are difficult described by traditional equations and have an enormous application potential.

As to the equations including real operators, engineering computation often need the approximate solutions reflecting an intuitive order relation and equivalence relation. Although the order relation and equivalence relation of real numbers are consistent, they are not as intuitive as those of base-b expansions. In practice, it is quicker to determine the order relation and equivalence relation of base-b expansions. So we introduce numerical computations to approximate real numbers with base-b expansions.

The numerical computations we proposed are not the best methods to approximate real numbers with base-b expansions, but the simple methods to approximate real numbers with base-b expansions. The computation complexity of the numerical computations we proposed could be promoted furtherly. However, we first prove that the Operator axioms can run on any modern computer. The numerical computation we proposed blends mathematics and computer science. Modern science depends on both the mathematics and the computer science. Arithmetic is the core of both the mathematics and the science. As a senior arithmetic, the Operator axioms will promote both the mathematics and the science in the future.

**Theorem 1.1.** *Any positive number  $\xi$  may be expressed as a limit of an infinite base-b expansion sequence*

$$\lim_{n \rightarrow \infty} A_1 A_2 \cdots A_{s+1} \cdot a_1 a_2 a_3 \cdots a_n, \quad (1.1)$$

where  $0 \leq A_1 < b, 0 \leq A_2 < b, \dots, 0 \leq a_n < b$ , not all  $A$  and  $a$  are 0, and an infinity of the  $a_n$  are less than (b-1). If  $\xi \geq 1$ , then  $A_1 \geq 0$ .

*Proof.* Let  $[\xi]$  be the integral part of  $\xi$ . Then we write

$$\xi = [\xi] + x = X + x, \quad (1.2)$$

where  $X$  is an integer and  $0 \leq x < 1$ , and consider  $X$  and  $x$  separately.

If  $X > 0$  and  $b^s \leq x < b^{s+1}$ , and  $A_1$  and  $X_1$  are the quotient and remainder when  $X$  is divided by  $b^s$ , then  $X = A_1 \cdot b^s + X_1$ , where  $0 < A_1 = [b^{-s}X] < b$ ,  $0 \leq X_1 < b^s$ .

Similarly

$$\begin{aligned} X_1 &= A_2 \cdot b^{s-1} + X_2 \quad (0 \leq A_2 < b, 0 \leq X_2 < b^{s-1}), \\ X_2 &= A_3 \cdot b^{s-2} + X_3 \quad (0 \leq A_3 < b, 0 \leq X_3 < b^{s-2}), \\ &\dots &&\dots \\ X_{s-1} &= A_s \cdot b + X_s \quad (0 \leq A_s < b, 0 \leq X_s < b), \\ X_s &= A_{s+1} \quad (0 \leq A_{s+1} < b). \end{aligned}$$

Thus  $X$  may be expressed uniquely in the form

$$X = A_1 \cdot b^s + A_2 \cdot b^{s-1} + \cdots + A_s \cdot b + A_{s+1}, \quad (1.3)$$

where every  $A$  is one of  $0, 1, \dots, (b-1)$ , and  $A_1$  is not 0. We abbreviate this expression to

$$X = A_1 A_2 \cdots A_s A_{s+1}, \quad (1.4)$$

the ordinary representation of  $X$  in base- $b$  expansion notation.

Passing to  $x$ , we write

$$X = f_1 \quad (0 \leq f_1 < 1).$$

We suppose that  $a_1 = [bf_1]$ , so that

$$\frac{a_1}{b} \leq f_1 < \frac{a_1 + 1}{b};$$

$a_1$  is one of  $0, 1, \dots, (b-1)$ , and

$$a_1 = [bf_1], \quad bf_1 = a_1 + f_2 \quad (0 \leq f_2 < 1).$$

Similarly, we define  $a_2, a_3, \dots$  by

$$a_2 = [bf_2], \quad bf_2 = a_2 + f_3 \quad (0 \leq f_3 < 1),$$

$$a_3 = [bf_3], \quad bf_3 = a_3 + f_4 \quad (0 \leq f_4 < 1),$$

...      ...      ...

Every  $a_n$  is one of  $0, 1, \dots, (b-1)$ . Thus

$$x = x_n + g_{n+1}, \quad (1.5)$$

where

$$x_n = \frac{a_1}{b} + \frac{a_2}{b^2} + \cdots + \frac{a_n}{b^n}, \quad (1.6)$$

$$0 \leq g_{n+1} = \frac{f_{n+1}}{b^n} < \frac{1}{b^n}. \quad (1.7)$$

We thus define a base- $b$  expansion  $.a_1 a_2 a_3 \cdots a_n \cdots$  associated with  $x$ . We call  $a_1, a_2, \dots$  the first, second,  $\dots$  digits of the base- $b$  expansion.

Since  $a_n < b$ , the series

$$\sum_1^{\infty} \frac{a_n}{b^n} \quad (1.8)$$

is convergent; and since  $g_{n+1} \rightarrow 0$ , its sum is  $x$ . We may therefore write

$$x = .a_1 a_2 a_3 \cdots, \quad (1.9)$$

the right-hand side being an abbreviation for the series (1.8).

We now combine (1.2), (1.4), and (1.9) in the form

$$\xi = X + x = A_1 A_2 \cdots A_s A_{s+1}. a_1 a_2 a_3 \cdots; \quad (1.10)$$

and the claim follows.  $\square$

Theorem 1.1 implies that every real number has base- $b$  expansions arbitrary close to it. So in numerical computations for the Operator axioms, all operands and outputs are denoted by base- $b$  expansions to intuitively show the order relation and equivalence relation.

The paper is organized as follows. In Section 2, we define the operation order for all operations in the Operator axioms. In Section 3, we construct the numerical computations for binary operations. In Section 4, we define some concepts in the Operator axioms.

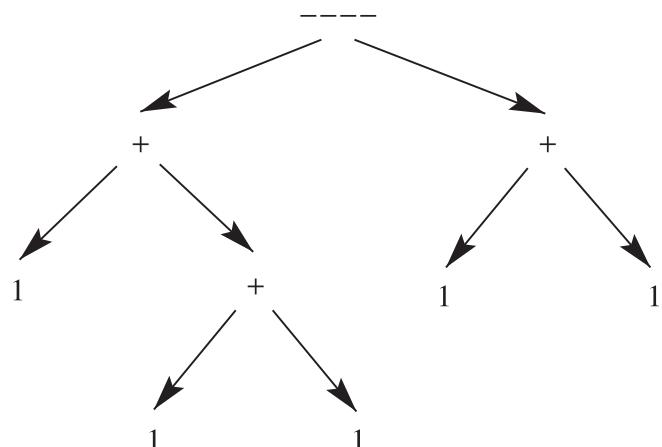
## 2. Operation Order

In the Operator axioms, the number ‘1’ is the only base- $b$  expansion while the others derive from the operation of two numbers and one operator. For example, the number “[1 + [1 + 1]] – – – [1 + 1]” derives from the operation of the number “[1 + [1 + 1]]”, the number “[1 + 1]” and the real operator “– – –”.

**Definition 2.1.** *Numerical computation* is a conversion from an operation to an approximate base- $b$  expansion.

In general, an operation includes many binary operators. For example, the operation “[1 + [1 + 1]] – – – [1 + 1]” includes three “+” and one “– – –”. Since each operator produces a binary operation,  $n$  operator in an operation will produce  $n$  binary operations. It is better to compute all binary operations in an operation in order. The order is denoted as *Operation Order*.

[4, §5.3.1] stores tradition operations as an expression tree and then applies traversal algorithm to evaluate the expression tree. Likewise, each operation of the Operator axioms can be stored as an expression tree in which each number ‘1’ become a leaf node and each operator become an internal node. Then Operation Order is just the traversal order of the expression tree. In this paper, we choose inorder traversal as Operation Order. Figure 1 illustrates an expression tree for the number “[1 + [1 + 1]] – – – [1 + 1]”.



**Figure 1.** An expression tree for the number “[1 + [1 + 1]] – – – [1 + 1]”.

It is supposed that the numerical computation applies base-10 expansions. Then the numerical computation for “[[1 + [1 + 1]] – – – [1 + 1]]” will proceed with the following Operation Order:

$$\begin{aligned} [[1 + [1 + 1]] - - [1 + 1]] &= [[1 + 2] - - - [1 + 1]] \\ &= [3 - - - [1 + 1]] \\ &= [3 - - - 2] \end{aligned}$$

In summary, every operation in the Operator axioms can divide into many binary operations by Operation Order. So numerical computations focus on the binary operations.

### 3. Numerical computations for binary operations

In [3], Operator axioms have expressed the real number system. In this section, “real number” refer to the real number deduced from Operator axioms [3].

#### 3.1. Division of binary operations

According to the complexity of numerical computations, we divide binary operations into low operations, middle operations and high operations. Table 1 lists their elements in detail.

**Table 1.** Division of binary operations.

	Low operations	Middle operations	High operations
Operators	$+, ++, -, --, /, //$	$+, --, //$	$++, +++, --, --, //, //, //, //$

#### 3.2. Numerical computations for low operations

In the Operator axioms,  $/$  is equal to  $-$  while  $//$  is equal to  $--$ . From a traditional viewpoint, the low operations “ $+, ++, -, --$ ” are equal to basic arithmetic operations “ $+, \times, -, \div$ ”. So numerical computations for low operations have been constructed in elementary arithmetic.

#### 3.3. Numerical computations for middle operations

From a traditional viewpoint,  $++$  is an exponentiation operation,  $--$  is a root-extraction operation and  $//$  is a logarithm operation. In this subsection, we import the numerical computations for the middle operations “ $++, --, //$ ” in [5, §23].

Let  $e$  be Euler’s number. It is supposed that  $a \in (-\infty, +\infty)$  is a base-b expansion,  $n \in \mathbb{Z}$  and  $k \in \mathbb{N}$ . The numerical computation for  $[e + + + a]$  can be constructed with the Taylor-series expansion as follows.

$$\begin{aligned} [e + + + a] &= \left[ \lim_{k \rightarrow +\infty} \left( \sum_{n=0}^k [[a + + + n] - - [n!]] \right) \right] \\ &\approx [1 + [a - - [1!]] + [[a + + + 2] - - [2!]] + [[a + + + 3] - - [3!]] + \dots + \end{aligned}$$

$$[[a + + + n] - -[n!]] + \dots$$

It is supposed that  $a \in (0, +\infty)$  is a base-b expansion and  $n \in \mathbb{Z}$ . Let  $b = [[a - 1] - -[a + 1]]$ , then the numerical computation for  $[a//e]$  can be constructed with the Taylor-series expansion as follows.

$$\begin{aligned} [a//e] &= [[[1 + b] - -[1 - b]]//e] \\ &= \left[ 2 + + \left[ \lim_{k \rightarrow +\infty} \left( \sum_{n=0}^k [[b + + + [2n + 1]] - -[2n + 1]] \right) \right] \right] \\ &\approx [2 + + [b + [[b + + + 3] - -3] + [[b + + + 5] - -5] + \dots + [[b + + + [2n + 1]] - -[2n + 1]] + \dots]]] \end{aligned}$$

It is supposed that  $a$  and  $b$  are two base-b expansions, where  $a \in (0, +\infty)$  and  $b \in (-\infty, +\infty)$ . Then the numerical computation for  $[a + + + b]$  can be divided and conquered with the identity  $[a + + + b] = [e + + + [b + + [a//e]]]$ .

It is supposed that  $[a + + + b]$  is a real number, where  $a \in (-\infty, 0]$  and  $b \in (-\infty, +\infty)$  are two base-b expansions. Then the numerical computation for  $[a + + + b]$  can always be equated with the basic numerical computations as above and the basic arithmetic operations in the axioms (OA.1)~(OA.75).

### 3.4. Numerical computations for high operations

It is supposed that the constants  $n, a_1, a_2, b_1, b_2, c_1, c_2, m \in \mathbb{N}$ .

Zero can be equated with a fraction  $[[1 - 1] - -1]$ . Any non-zero base-b expansion can be equated with a fraction  $[[1 - 1] \pm [a_1 - -a_2]]$  for  $a_1, a_2 \in \mathbb{N}$ .

It is supposed that the constant  $d \in \mathbb{R}$  with  $[1 - 1] < d$ . It is supposed that the constant  $e \in \mathbb{R}$  with  $1 < e$ . For clarity, we rename VE Function [3] to Prefix Function. For clarity, we rename EV Function [3] to Suffix Function.

**Definition 3.1.** [3, Definition 3.1] *Prefix Function* is the function  $f : [1, +\infty) \rightarrow \mathbb{R}$  defined by  $f(x) = [x + + + \dot{e}d]$ .

**Definition 3.2.** [3, Definition 3.2] *Suffix Function* is the function  $f : \mathbb{R} \rightarrow \mathbb{R}$  defined by  $f(x) = [e + + + \dot{e}x]$ .

**Definition 3.3.** [3, Definition 3.3] *Fundamental operator functions* are *Prefix Function* and *Suffix Function*.

**Theorem 3.4.** [3, Theorem 3.4] *The Prefix Function  $f(x) = [x + + + \dot{e}d]$  is continuous, unbounded and strictly increasing.*

**Theorem 3.5.** [3, Theorem 3.5] *The Suffix Function  $f(x) = [e + + + \dot{e}x]$  is continuous, unbounded and strictly increasing.*

It is supposed that the constant  $t, u, v \in \mathbb{R}$ .

**Definition 3.6.** *Root Equations* are the equations such as  $f(x) = t$  such that:

1. The function  $f(x)$  is real and continuous on any closed interval  $[u, v]$  in the domain;
2. The equation  $f(x) = t$  has only one root on the above interval  $[u, v]$ ;

[6, TABLE PT2.3] lists common root-finding methods and their convergence conditions. When  $a$  acts as the lower guess and  $b$  acts as the upper guess, both the bisection method [6, §5.2] and Brent's method [6, §6.4] always converge and find the only root on  $[u, v]$ . But Brent's method converges faster than the bisection method and thus acts as the main root-finding method for Root Equations.

In the following, we construct all numerical computations for high operations by induction. It is supposed that the constant  $p_1 \in R$  with  $1 \leq p_1$ . It is supposed that the constant  $p_2 \in R$ . It is supposed that the constant  $q_1 \in R$  with  $1 \leq q_1$ . It is supposed that the constant  $q_2 \in R$  with  $[1 - 1] \leq q_2$ . It is supposed that the constant  $r_1 \in R$  with  $[1 - 1] < r_1$ . It is supposed that the constant  $r_2 \in R$  with  $1 < r_2$ . In the following, we approximate these constants with those fractions such as  $[a_1 -- a_2]$ .

1. The numerical computations for the operations  $[p_1 + + + + p_2]$ ,  $[q_1 -- - q_2]$ ,  $[r_1 // / r_2]$  are constructed.

(a) A numerical computation for  $[[a_1 -- a_2] + + + + [b_1 -- b_2]]$  with  $1 \leq [a_1 -- a_2]$  and  $[1 - 1] \leq [b_1 -- b_2]$  can be constructed.

i.  $[b_1 -- b_2] \leq 1$ .

$$\begin{aligned}
 (A1) \quad & (\bar{c} = [[[a_1 -- a_2] - 1] + + [1 + [1 + 1]]]) \wedge \\
 & (\bar{d} = [[[a_1 -- a_2] - 1] + + [1 + 1]]) \wedge \\
 & (\bar{e} = [[b_1 -- b_2] + + [1 + 1]]) \wedge \\
 & (\bar{f} = [[b_1 -- b_2] + + [1 + [1 + 1]]]) \wedge \\
 & ([[a_1 -- a_2] + + + + [b_1 -- b_2]] = \\
 & [[1 + [\bar{c} + + + + 1]]] - [\bar{d} + + + + [\bar{f} + + + + 1]]]) \quad \text{by (OA.83),(OA.28)}
 \end{aligned}$$

$$\begin{aligned}
 (A2) \quad & (\bar{c} = [[[a_1 -- a_2] - 1] + + [1 + [1 + 1]]]) \wedge \\
 & (\bar{d} = [[[a_1 -- a_2] - 1] + + [1 + 1]]) \wedge \\
 & (\bar{e} = [[b_1 -- b_2] + + + [1 + 1]]) \wedge \\
 & (\bar{f} = [[b_1 -- b_2] + + + [1 + [1 + 1]]]) \wedge \\
 & ([[a_1 -- a_2] + + + + [b_1 -- b_2]] = \\
 & [[1 + [\bar{c} + + + + 1]]] - [\bar{d} + + + + [\bar{f} + + + + 1]]]) \quad \text{by (OA.83),(OA.62)}
 \end{aligned}$$

ii.  $1 < [b_1 -- b_2]$ .

$$\begin{aligned}
 (B1) \quad & [[a_1 -- a_2] + + + + [b_1 -- b_2]] = \\
 & [[a_1 -- a_2] + + + [[a_1 -- a_2] + + + + [[b_1 -- b_2] - 1]]] \quad \text{by (OA.82)}
 \end{aligned}$$

(B2) It is supposed that  $[1 - 1] \leq [[b_1 -- b_2] - n]$  and  
 $[[b_1 -- b_2] - n] \leq 1$ . Let us distinguish these  
 $[a_1 -- a_2]$  with the subscripts  $\{(1), (2), (3), \dots\}$ .

$$\begin{aligned}
 (B3) \quad & \Rightarrow [[a_1 -- a_2] + + + + [b_1 -- b_2]] = \\
 & [[a_1 -- a_2]_{(1)} + + + [[a_1 -- a_2]_{(2)} + + + \dots] \\
 & [[a_1 -- a_2]_n + + + [[a_1 -- a_2] + + + + \dots]
 \end{aligned}$$

$$\begin{aligned}
& [[b_1 - -b_2] - n] \cdots ] \quad \text{by (OA.82),(B2)} \\
(B4) \quad & (\bar{c} = [[[a_1 - -a_2] - 1] + +[1 + [1 + 1]]]) \wedge \\
& (\bar{d} = [[[a_1 - -a_2] - 1] + +[1 + 1]])) \wedge \\
& (\bar{e} = [[[b_1 - -b_2] - n] + + + [1 + 1]])) \wedge \\
& (\bar{f} = [[[b_1 - -b_2] - n] + + + [1 + [1 + 1]]]) \wedge \\
& ([[a_1 - -a_2] + + + [[b_1 - -b_2] - n]] = \\
& [[1 + [\bar{c} + +\bar{e}]] - [\bar{d} + +\bar{f}]]]) \quad \text{by (B2),(A2)} \\
(B5) \quad & \Rightarrow (\bar{c} = [[[a_1 - -a_2] - 1] + +[1 + [1 + 1]]]) \wedge \\
& (\bar{d} = [[[a_1 - -a_2] - 1] + +[1 + 1]])) \wedge \\
& (\bar{e} = [[[b_1 - -b_2] - n] + + + [1 + 1]])) \wedge \\
& (\bar{f} = [[[b_1 - -b_2] - n] + + + [1 + [1 + 1]]]) \wedge \\
& ([[a_1 - -a_2] + + + [b_1 - -b_2]] = \\
& [[a_1 - -a_2]_{(1)} + + + [[a_1 - -a_2]_{(2)} + + + \cdots \\
& [[a_1 - -a_2]_n + + + [[1 + [\bar{c} + +\bar{e}]] - [\bar{d} + +\bar{f}]] \cdots ]]) \quad \text{by (B3),(B4)}
\end{aligned}$$

Then (A1)~(A2) and (B1)~(B4) have reduced one  $+++$  operation to many  $+++$  operations. Since §3.3 has constructed the numerical computation for  $[[a_1 - -a_2] + + + [b_1 - -b_2]]$ , (A1)~(A2) and (B1)~(B4) can achieve a numerical computation for  $[[a_1 - -a_2] + + + [b_1 - -b_2]]$ .

(b) A numerical computation for  $[[a_1 - -a_2] - - - [b_1 - -b_2]]$  with  $1 \leq [a_1 - -a_2]$  and  $[1 - 1] < [b_1 - -b_2]$  can be constructed.

The numerical computation for  $[[a_1 - -a_2] - - - [b_1 - -b_2]]$  is equated with the numerical root finding of the equation  $x = [[a_1 - -a_2] - - - [b_1 - -b_2]]$ .

$$\begin{aligned}
(A1) \quad & x = [[a_1 - -a_2] - - - [b_1 - -b_2]] \\
(A2) \quad & \Rightarrow [x + + + [b_1 - -b_2]] = \\
& [[a_1 - -a_2] - - - [b_1 - -b_2]] \\
& + + + [b_1 - -b_2]] \quad \text{by (OA.108)} \\
(A3) \quad & \Rightarrow [x + + + [b_1 - -b_2]] = [a_1 - -a_2] \quad \text{by (OA.79),(OA.24),} \\
& \quad \text{(OA.25)}
\end{aligned}$$

According to (OA.37)~(OA.40), the function  $f(x) = [x + + + [b_1 - -b_2]]$  is defined on the domain  $[1, +\infty)$ . Theorem 3.4 implies that we can iteratively increase  $v$  by step 1 from  $v = 1$  until  $[a_1 - -a_2] < [v + + + [b_1 - -b_2]]$  holds.

Theorem 3.4 implies that  $f(x) = [x + + + [b_1 - -b_2]]$  is continuous on the domain  $[1, v]$ . (OA.75) derives that  $[1 + + + [b_1 - -b_2]] = 1$ . So (OA.95) derives that  $[1 + + + [b_1 - -b_2]] \leq [a_1 - -a_2]$ . In summary, both  $[1 + + + [b_1 - -b_2]] \leq [a_1 - -a_2]$  and  $[a_1 - -a_2] < [v + + + [b_1 - -b_2]]$  hold.

Then Intermediate Value Theorem derives that the equation  $[x + + + [b_1 - -b_2]] = [a_1 - -a_2]$  has only one root on the domain  $[1, v]$ . Since Theorem 3.4 implies that the equation

$[x + + + + [b_1 - - b_2]] = [a_1 - - a_2]$  has no root on the domain  $(v, +\infty)$ , the equation  $[x + + + + [b_1 - - b_2]] = [a_1 - - a_2]$  has only one root on the domain  $[[1 - 1], +\infty)$ . Since the equation  $[x + + + + [b_1 - - b_2]] = [a_1 - - a_2]$  belongs to Root Equations, Brent's method can find the only root of the equation and constructs the numerical computation for  $[[a_1 - - a_2] - - - [b_1 - - b_2]]$ .

(c) A numerical computation for  $[[a_1 - - a_2] // / / [b_1 - - b_2]]$  with  $1 \leq [a_1 - - a_2]$  and with  $1 < [b_1 - - b_2]$  can be constructed.

The numerical computation for  $[[a_1 - - a_2] // / / [b_1 - - b_2]]$  is equated with the numerical root finding of the equation  $x = [[a_1 - - a_2] // / / [b_1 - - b_2]]$ .

$$\begin{aligned}
 (A1) \quad & x = [[a_1 - - a_2] // / / [b_1 - - b_2]] \\
 (A2) \quad & [1 - 1] < [[a_1 - - a_2] // / / [b_1 - - b_2]] \quad \text{by (OA.33)} \\
 (A3) \quad & \Rightarrow 1 < [[b_1 - - b_2] + + + + \\
 & [[a_1 - - a_2] // / / [b_1 - - b_2]]] \quad \text{by (OA.37),(OA.72)} \\
 (A4) \quad & \Rightarrow ([[b_1 - - b_2] + + + + \\
 & [[a_1 - - a_2] // / / [b_1 - - b_2]]]) \quad \text{by (OA.90)} \\
 (A5) \quad & \Rightarrow [[b_1 - - b_2] + + + + x] = \\
 & [[b_1 - - b_2] + + + + \\
 & [[a_1 - - a_2] // / / [b_1 - - b_2]]] \quad \text{by (OA.105)} \\
 (A6) \quad & \Rightarrow [[b_1 - - b_2] + + + + x] = [a_1 - - a_2] \quad \text{by (OA.80),(OA.104)}
 \end{aligned}$$

According to (OA.37)~(OA.40) and (OA.73), the function  $f(x) = [[b_1 - - b_2] + + + + x]$  is defined on the domain  $(-\infty, +\infty)$ . Theorem 3.5 implies that we can iteratively decrease  $u$  by step 1 from  $u = 1$  until  $[[b_1 - - b_2] + + + + u] < [a_1 - - a_2]$  holds. Theorem 3.5 also implies that we can iteratively increase  $v$  by step 1 from  $v = 1$  until  $[a_1 - - a_2] < [[b_1 - - b_2] + + + + v]$  holds.

Theorem 3.5 implies that  $f(x) = [[b_1 - - b_2] + + + + x]$  is continuous on the domain  $(-\infty, +\infty)$ . In summary, both  $[[b_1 - - b_2] + + + + u] < [a_1 - - a_2]$  and  $[a_1 - - a_2] < [[b_1 - - b_2] + + + + v]$  hold.

Then Intermediate Value Theorem derives that the equation  $[[b_1 - - b_2] + + + + x] = [a_1 - - a_2]$  has only one root on the domain  $[u, v]$ . Since Theorem 3.5 implies that the equation  $[[b_1 - - b_2] + + + + x] = [a_1 - - a_2]$  has no root on the domains  $(-\infty, u)$  and  $(v, +\infty)$ , the equation  $[[b_1 - - b_2] + + + + x] = [a_1 - - a_2]$  has only one root on the domain  $[u, v]$ . Since the equation  $[[b_1 - - b_2] + + + + x] = [a_1 - - a_2]$  belongs to Root Equations, Brent's method can find the only root of the equation and constructs the numerical computation for  $[[a_1 - - a_2] // / / [b_1 - - b_2]]$ .

2. If the numerical computations for  $[p_1 + + + \dot{e}p_2]$ ,  $[q_1 - - - \dot{f}q_2]$ ,  $[r_1 // / \dot{g}r_2]$  have been constructed, then the numerical computations for  $[p_1 + + + + \dot{e}p_2]$ ,  $[q_1 - - - - \dot{f}q_2]$ ,  $[r_1 // / / \dot{g}r_2]$  can also be constructed.

According to (OA.19), the symbol ‘ $e$ ’ represents some successive ‘ $+$ —“ $+ \cdots +$ ”. According to (OA.20), the symbol ‘ $f$ ’ represents some successive ‘ $-$ —“ $- \cdots -$ ”. According to (OA.21), the symbol ‘ $g$ ’ represents some successive ‘ $/$ —“ $/ \cdots /$ ”.

(a) A numerical computation for  $[[a_1 - - a_2] + + + + \dot{e}[b_1 - - b_2]]$  with  $1 \leq [a_1 - - a_2]$  and  $[1 - 1] \leq [b_1 - - b_2]$  can be constructed.

i.  $[b_1 - -b_2] \leq 1$ .

$$\begin{aligned}
 (A1) \quad & (\bar{c} = [[[a_1 - -a_2] - 1] + +[1 + [1 + 1]]]) \wedge \\
 & (\bar{d} = [[[a_1 - -a_2] - 1] + +[1 + 1]]) \wedge \\
 & (\bar{e} = [[b_1 - -b_2] + + + [1 + 1]]) \wedge \\
 & (\bar{f} = [[b_1 - -b_2] + + + [1 + [1 + 1]]]) \wedge \\
 & ([[a_1 - -a_2] + + + + \dot{h}[b_1 - -b_2]] = \\
 & [[1 + [\bar{c} + + + [\bar{e} + + + [1 + \dot{k}]]]]] - \\
 & [\bar{d} + + + [\bar{f} + + + [1 + \dot{k}]]]) \quad \text{by (OA.83),(OA.29)}
 \end{aligned}$$

ii.  $1 < [b_1 - -b_2]$ .

$$\begin{aligned}
 (B1) \quad & [[a_1 - -a_2] + + + + \dot{e}[b_1 - -b_2]] = \\
 & [[a_1 - -a_2] + + + \dot{e}[[a_1 - -a_2] + + + + \dot{e}[[b_1 - -b_2] - 1]]]] \quad \text{by (OA.82)} \\
 (B2) \quad & \text{It is supposed that } [1 - 1] \leq [[b_1 - -b_2] - n] \text{ and} \\
 & [[b_1 - -b_2] - n] \leq 1. \text{ Let us distinguish these} \\
 & [a_1 - -a_2] \text{ with the subscripts } \{(1), (2), (3), \dots\}. \\
 (B3) \quad & \Rightarrow [[a_1 - -a_2] + + + + \dot{e}[b_1 - -b_2]] = \\
 & [[a_1 - -a_2]_{(1)} + + + \dot{e}[[a_1 - -a_2]_{(2)} + + + + \dot{e} \dots \\
 & [[a_1 - -a_2]_n + + + + \dot{e}[[a_1 - -a_2] + + + + \dot{e} \\
 & [[b_1 - -b_2] - n]] \dots]]] \quad \text{by (OA.82),(B2)} \\
 (B4) \quad & (\bar{c} = [[[a_1 - -a_2] - 1] + +[1 + [1 + 1]]]) \wedge \\
 & (\bar{d} = [[[a_1 - -a_2] - 1] + +[1 + 1]]) \wedge \\
 & (\bar{e} = [[[b_1 - -b_2] - n] + + + [1 + 1]]) \wedge \\
 & (\bar{f} = [[[b_1 - -b_2] - n] + + + [1 + [1 + 1]]]) \wedge \\
 & ([[a_1 - -a_2] + + + + \dot{e}[[b_1 - -b_2] - n]] = \\
 & [[1 + [\bar{c} + + \bar{e}]] - [\bar{d} + + \bar{f}]]]) \quad \text{by (B3),(A1)} \\
 (B5) \quad & \Rightarrow (\bar{c} = [[[a_1 - -a_2] - 1] + +[1 + [1 + 1]]]) \wedge \\
 & (\bar{d} = [[[a_1 - -a_2] - 1] + +[1 + 1]]) \wedge \\
 & (\bar{e} = [[[b_1 - -b_2] - n] + + + [1 + 1]]) \wedge \\
 & (\bar{f} = [[[b_1 - -b_2] - n] + + + [1 + [1 + 1]]]) \wedge \\
 & ([[a_1 - -a_2] + + + + \dot{e}[b_1 - -b_2]] = \\
 & [[a_1 - -a_2]_{(1)} + + + + \dot{e}[[a_1 - -a_2]_{(2)} + + + + \dot{e} \dots \\
 & [[a_1 - -a_2]_n + + + + \dot{e}[[1 + [\bar{c} + + \bar{e}]] - [\bar{d} + + \bar{f}]] \dots]]]) \quad \text{by (B3),(B4)}
 \end{aligned}$$

Then (A1) and (B1)~(B5) have reduced one  $+++ + \dot{e}$  operation to many  $++ + \dot{e}$  operations. Since the numerical computation for  $[[a_1 - -a_2] + + + + \dot{e}[b_1 - -b_2]]$  has been

supposed to be constructed, (A1) and (B1)~(B5) can achieve a numerical computation for  $[[a_1 - -a_2] + + + + \dot{e}[b_1 - -b_2]]$ .

(b) A numerical computation for  $[[a_1 - -a_2] - - - \dot{f}[b_1 - -b_2]]$  with  $1 \leq [a_1 - -a_2]$  and  $[1 - 1] < [b_1 - -b_2]$  can be constructed.

The numerical computation for  $[[a_1 - -a_2] - - - \dot{f}[b_1 - -b_2]]$  is equated with the numerical root finding of the equation  $x = [[a_1 - -a_2] - - - \dot{f}[b_1 - -b_2]]$ .

$$(A1) \quad x = [[a_1 - -a_2] - - - \dot{f}[b_1 - -b_2]]$$

$$(A2) \quad \Rightarrow [x + + + + \dot{e}[b_1 - -b_2]] =$$

$$[[[a_1 - -a_2] - - - - i[b_1 - -b_2]]$$

$$+ + + + h[b_1 - -b_2]] \quad \text{by (OA.105)}$$

$$(A3) \quad \Rightarrow [x + + + + \dot{e}[b_1 - -b_2]] = [a_1 - -a_2] \quad \text{by (OA.78),(OA.24),} \\ \text{(OA.25)}$$

According to (OA.37)~(OA.40), the function  $f(x) = [x + + + + \dot{e}[b_1 - -b_2]]$  is defined on the domain  $[1, +\infty)$ . Theorem 3.4 implies that we can iteratively increase  $v$  by step 1 from  $v = 1$  until  $[a_1 - -a_2] < [v + + + + \dot{e}[b_1 - -b_2]]$  holds.

Theorem 3.4 implies that  $f(x) = [x + + + + \dot{e}[b_1 - -b_2]]$  is continuous on the domain  $[1, v]$ . (OA.75) derives that  $[1 + + + + \dot{e}[b_1 - -b_2]] = 1$ . So (OA.95) derives that  $[1 + + + + \dot{e}[b_1 - -b_2]] \leq [a_1 - -a_2]$ . In summary, both  $[1 + + + + \dot{e}[b_1 - -b_2]] \leq [a_1 - -a_2]$  and  $[a_1 - -a_2] < [v + + + + \dot{e}[b_1 - -b_2]]$  hold.

Then Intermediate Value Theorem derives that the equation  $[x + + + + \dot{e}[b_1 - -b_2]] = [a_1 - -a_2]$  has only one root on the domain  $[1, v]$ . Since Theorem 3.4 implies that the equation  $[x + + + + \dot{e}[b_1 - -b_2]] = [a_1 - -a_2]$  has no root on the domain  $(v, +\infty)$ , the equation  $[x + + + + \dot{e}[b_1 - -b_2]] = [a_1 - -a_2]$  has only one root on the domain  $[[1 - 1], +\infty)$ . Since the equation  $[x + + + + \dot{e}[b_1 - -b_2]] = [a_1 - -a_2]$  belongs to Root Equations, Brent's method can find the only root of the equation and constructs the numerical computation for  $[[a_1 - -a_2] - - - \dot{f}[b_1 - -b_2]]$ .

(c) A numerical computation for  $[[a_1 - -a_2] // / / \dot{g}[b_1 - -b_2]]$  with  $1 \leq [a_1 - -a_2]$  and with  $1 < [b_1 - -b_2]$  can be constructed.

The numerical computation for  $[[a_1 - -a_2] // / / \dot{g}[b_1 - -b_2]]$  is equated with the numerical root finding of the equation  $x = [[a_1 - -a_2] // / / \dot{g}[b_1 - -b_2]]$ .

$$(A1) \quad x = [[a_1 - -a_2] // / / \dot{g}[b_1 - -b_2]]$$

$$(A2) \quad [1 - 1] < [[a_1 - -a_2] // / / \dot{g}[b_1 - -b_2]] \quad \text{by (OA.33)}$$

$$(A3) \quad \Rightarrow 1 < [[b_1 - -b_2] + + + + h$$

$$[[a_1 - -a_2] // / / j[b_1 - -b_2]]] \quad \text{by (OA.37),(OA.72)}$$

$$(A4) \quad \Rightarrow ([[b_1 - -b_2] + + + + h$$

$$[[a_1 - -a_2] // / / j[b_1 - -b_2]]]) \quad \text{by (OA.90)}$$

$$(A5) \quad \Rightarrow [[b_1 - -b_2] + + + + \dot{e}x] =$$

$$[[b_1 - -b_2] + + + + h$$

$$[[a_1 - -a_2] // / / j[b_1 - -b_2]]] \quad \text{by (OA.105)}$$

$$(A6) \Rightarrow [[b_1 -- b_2] + + + + \dot{e}x] = [a_1 -- a_2] \text{ by (OA.80),(OA.104)}$$

According to (OA.37)~(OA.40) and (OA.73), the function  $f(x) = [[b_1 -- b_2] + + + + \dot{e}x]$  is defined on the domain  $(-\infty, +\infty)$ . Theorem 3.5 implies that we can iteratively decrease  $u$  by step 1 from  $u = 1$  until  $[[b_1 -- b_2] + + + + \dot{e}u] < [a_1 -- a_2]$  holds. Theorem 3.5 also implies that we can iteratively increase  $v$  by step 1 from  $v = 1$  until  $[a_1 -- a_2] < [[b_1 -- b_2] + + + + \dot{e}v]$  holds.

Theorem 3.5 implies that  $f(x) = [[b_1 -- b_2] + + + + \dot{e}x]$  is continuous on the domain  $(-\infty, +\infty)$ . In summary, both  $[[b_1 -- b_2] + + + + \dot{e}u] < [a_1 -- a_2]$  and  $[a_1 -- a_2] < [[b_1 -- b_2] + + + + \dot{e}v]$  hold.

Then Intermediate Value Theorem derives that the equation  $[[b_1 -- b_2] + + + + \dot{e}x] = [a_1 -- a_2]$  has only one root on the domain  $[u, v]$ . Since Theorem 3.5 implies that the equation  $[[b_1 -- b_2] + + + + \dot{e}x] = [a_1 -- a_2]$  has no root on the domains  $(-\infty, u)$  and  $(v, +\infty)$ , the equation  $[[b_1 -- b_2] + + + + \dot{e}x] = [a_1 -- a_2]$  has only one root on the domain  $[u, v]$ . Since the equation  $[[b_1 -- b_2] + + + + \dot{e}x] = [a_1 -- a_2]$  belongs to Root Equations, Brent's method can find the only root of the equation and constructs the numerical computation for  $[[a_1 -- a_2] // / \dot{g}[b_1 -- b_2]]$ .

3. By induction, the numerical computations for  $[p_1+++++p_2]$ ,  $[p_1++++++p_2]$ ,  $[p_1++++++p_2]$ ,  $\dots$ ,  $[q_1 - - - - - q_2]$ ,  $[q_1 - - - - - q_2]$ ,  $[q_1 - - - - - q_2]$ ,  $\dots$ ,  $[r_1 // / / r_2]$ ,  $[r_1 // / / r_2]$ ,  $[r_1 // / / / r_2]$ ,  $\dots$  can all be constructed.

#### 4. Some concepts in the Operator axioms

We define some replacements for the notations of the Operator axioms, as is shown in Table 2.

The notation  $+'$  can be replaced by any element of the set  $\{+, ++, + + +, + + + +, \dots\}$ . The notation  $-'$  can be replaced by any element of the set  $\{-, --, ---, ----, \dots\}$ . The notation  $/'$  can be replaced by any element of the set  $\{/, //, // /, // / /, \dots\}$ .

We define the pronunciations for some expressions in the Operator axioms, as is shown in Table 3.

We divide the real operators into an ordered level with the natural numbers. Table 4 lists the levels of the real operators in detail.

The order of real operators is listed as follows:

level-1 < level-2 < level-3 <  $\dots$  < level-n

We define the operations of real operators as follows.

**Definition 4.1.** *Complete Operations* are all the binary operations of real operators.

According to the Definition 4.1, all operations such as  $a +'_n b$ ,  $a '-'_n b$ ,  $a /'_n b$  compose the complete operations.

**Table 2.** Replacements of notations.

Replacement	Notation in Operator axioms
$+'$	$\dot{e}$
$-'$	$\dot{f}$
$/'$	$\dot{j}$
$+'_1$	$+$
$+'_2$	$++$
$+'_3$	$+++$
$+'_n$	$\underbrace{++\cdots+}_n$
$-'_1$	$-$
$-'_2$	$--$
$-'_3$	$-- -$
$-'_n$	$\underbrace{--\cdots-}_n$
$/'_1$	$/$
$/'_2$	$//$
$/'_3$	$///$
$/'_n$	$\underbrace{///\cdots/}_n$

**Table 3.** Pronunciations for some expressions.

Expression	Pronunciation
$+'$	addote
$-'$	subote
$/'$	logote
$a +'_n b$	a addote n to b
$a -'_n b$	a subote n to b
$a /'_n b$	a logote n to b

**Table 4.** Level of operators.

Level	Operators
1	$+'_1, -'_1, /'_1$
2	$+'_2, -'_2, /'_2$
3	$+'_3, -'_3, /'_3$
...	...
n	$+'_n, -'_n, /'_n$

---

## Conflict of interest

The author declares no conflict of interest.

## References

1. P. Xie, A logical calculus to intuitively and logically denote number systems, *Prog. Appl. Math.*, **1** (2011), 43–70.
2. P. Xie, Number systems based on logical calculus, *Int. Math. Forum*, **8** (2013), 1663–1689.
3. P. Xie, Improvement on operator axioms and fundamental operator functions, *Int. J. Math. Appl.*, **5** (2017), 125–151.
4. C. A. Shaffer, *Data structures and algorithm analysis in C++*, Third Edition, Dover Publications, 2011.
5. B. Parhami, *Computer arithmetic: algorithms and Hardware designs*, Oxford University Press, 2000.
6. S. C. Chapra, R. P. Canale, *Numerical methods for engineers*, Sixth Edition, McGraw-Hill Companies, Inc., 2009.



AIMS Press

© 2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)