Article

Not peer-reviewed version

On the Unary Function Clone Problem via Parametrized Arithmetic Functionals

Humphrey Takunda Muchapireyi *

Posted Date: 25 July 2025

doi: 10.20944/preprints202507.2176.v1

Keywords: Clone Theory; Minimal Unary Basis; PRA; Kalmar Class

Article

On the Unary Function Clone Problem via Parametrized Arithmetic Functionals

Humphrey Takunda Muchapireyi

Independent Researcher; humphreymuchapireyi1@gmail.com; Tel.: +263783721727

Abstract

Attempts to solve the Unary Function Clone Problem in Clone Theory have primarily focused on classifying unary fragments within primitive recursive and elementary function hierarchies. Conventional methods have considered closure properties, growth rates, and function arity. This treatise introduces an original, parameterized framework that formulates the elementary arithmetic operations as members of a unified unary family of arity-indexed functionals. Our theory is consistent with Kalmar's class of elementary functions, Peano Arithmetic, categorical-Lawvere and Gödelian semantics. Utilizing a single iteration parameter system, we show that these operations, though algebraically expressible, are clone-theoretically independent. We prove a minimal unary basis capable of generating the full arithmetic clone under composition.

Keywords: Clone Theory; Minimal Unary Basis; PRA; Kalmar Class

---

1. Introduction

In mathematical logic, recursion theory, and universal algebra, the pursuit of generative foundations of arithmetic operations has been a top priority. Particularly in clone theory—a framework that studies the closure of functions under composition—the question of whether the full suite of elementary arithmetic operations can be reconstructed from a minimal basis of simpler functions is equally central. A notorious open problem in this space is the Unary Function Clone Problem, which entails, 'can a clone of arithmetic functions can be generated using only unary functions, without relying on binary or higher-arity inputs.' Historical approaches to this problem have primarily focused on the use of function algebras, bounded recursion, or categorical semantics (e.g., via Lawvere theories), with notable results having been accomplished in understanding clones of elementary functions through Kalmar's classification [1] and primitive recursion frameworks. However, none have conclusively identified a minimal, closed, and clone-independent [2] unary basis that is sufficient for reconstructing elementary arithmetic. Some theories even stress the necessity of binary constructs, while others attempt to reduce function arities through diagonalization or encoding, often at the cost of generality. Clone theory has seen notable advances in the classification of minimal clones, particularly those generated by binary idempotent operations. Behrisch and Machida [7], analyzed the minimality of certain binary clones in the lattice of clones on finite domains using directed graphs and algebraic generators to investigate how binary functions relate to minimal clones, offering structural insights into clone minimality from a lattice-theoretic standpoint. This paper however introduces a candidate solution to the Function Clone Problem using a parameterized family Ƒ of unary functionals indexed by an integer. The functionals, for generalized addition, subtraction, multiplication, and division operations, are all defined over a single operand and closed under composition. The principal aim of this study is to construct and validate a minimal unary clone basis sufficient to generate all elementary arithmetic operations. We show that this system satisfies the conditions of closure, clone-theoretic independence, and constructability within Primitive Recursive Arithmetic (PRA), and also demonstrate its compatibility with Kalmar's bounded recursion, Lawvere categories, and Gödelian coding. Our study offers both

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2025 doi:10.20944/preprints202507.2176.v1

theoretical and computational insights, with potential applications in the foundations of arithmetic computation, Generative AI symbolic reasoning and formal methods.

2. Background

2.1. Clone Theory

A clone $\mathbf{C}$ on a set X is a collection of finitary operations $f: X^n \to X$ (for all n) such that:

- Closure under composition: If $f \in \mathbf{C}$ is n-ary and $g_1, \ldots, g_n \in \mathbf{C}$ are m-ary, then $f(g_1, \ldots, g_n \in \mathbf{C})$.
- Containment of projections: For every n, the projection $\pi^n(x_1, \ldots, x_n) = x_k$ is in $\mathbf{C}$.

The Unary Clone Basis Problem states: is there a finite set of unary functions on whose clone equals the clone of elementary or primitive recursive functions? [2].

2.2. Kalmar's Elementary Functions

Kalmar defined the class of elementary functions as those built from:

- Zero function: $Z(x) = 0$
- Successor: $S(x) = x + 1$
- Projection: $\pi_k{}^n$
- Composition
- Bounded sum and bounded product

Addition and multiplication are primitive in this system; exponentiation is not [1,3].

2.3. Primitive Recursive Arithmetic (PRA)

PRA is a fragment of arithmetic where functions are defined by:

- Initial functions: zero, successor, projection
- Closure under composition and primitive recursion

Addition, multiplication, and exponentiation (via iteration of multiplication) are definable. Division and subtraction are partial, or defined with guards to stay within N [4,6]. Formally, PRA allows one to define a function Ƒ via:

$$f\,(0) = g\,(x) \tag{1}$$

$$f\{s(n)\} = h\{n, f(n)\} \tag{2}$$

where g and h are already defined functions. This forms the core of primitive recursion. Bounded recursion schemes are also admissible under the schema:

$$f(n) = \begin{cases} \mathrm{g(n)} & \text{if } n < b \\ h(n, f(n-1)) & \textit{otherwise} \end{cases} \tag{3}$$

This structure supports constructive encoding of all elementary functions via bounded growth.

3. Construction

3.1. Unary Parametrized Arithmetic Operations

In our theory, we define a free operand $A \in \mathbf{N}$. Next, we propose novel unary functions that are parameterized by an index $i \in \mathbf{Z}$, a parameter controlling the operator's shape. We therefore define the following functions:

The Unary Addition Function:

$$add^i(A) = \sum_{j=1}^{1+i} A_j = A(1+i) \tag{4}$$

The Unary Subtraction Function:

$$\boldsymbol{sub^i}(\boldsymbol{A}) = \frac{1-i}{1+i}\sum_{j=1}^{1+i} \boldsymbol{A_j} = \boldsymbol{A}(\mathbf{1}-\boldsymbol{i}) \quad (5)$$

The Unary Multiplication Function:

$$\boldsymbol{mul^i}(\boldsymbol{A}) = \prod_{j=1}^{1+i} \boldsymbol{A_j} = \boldsymbol{A}^{\mathbf{1}+\boldsymbol{i}} \quad (6)$$

The Unary Division Function:

$$\boldsymbol{div^i}(\boldsymbol{A}) = \frac{\boldsymbol{A}^2}{\prod_{j=1}^{1+i} \boldsymbol{A_j}} = \boldsymbol{A}^{\mathbf{1}-\boldsymbol{i}} \quad (7)$$

In this study, we let the index belong the set of integers in order to enable the definition of dual operations (e.g., subtraction and division) alongside conventional addition and multiplication. While this traverses beyond the confines of primitive recursion over

N, it preserves closure under clone composition and allows a unified arithmetic structure. This extension is intentional and forms the basis for a minimal unary clone over parametrized arithmetic operations. The value of $\boldsymbol{i}$ encodes iteration-based views of the arithmetic operations such as: identity form, canonical arithmetic operation, iterative composition i.e., repetition and inverse relations. These functions define unary operations that generalize iteration depth or growth rate by $\boldsymbol{i}$. We now analyze their formal basis.

3.2. Peano Arithmetic Axioms

We interpret our unary function family through standard recursive definitions from Peano Arithmetic [4]. Peano Arithmetic defines arithmetic from axioms about 0, the successor function $\boldsymbol{S}(\boldsymbol{x})$, and induction:

1. $\forall \boldsymbol{x} \neg(\boldsymbol{S}(\boldsymbol{x}) = \mathbf{0}$
2. $\forall \boldsymbol{x}, \boldsymbol{y}(\boldsymbol{S}(\boldsymbol{x}) = \boldsymbol{S}(\boldsymbol{y}) \Rightarrow \boldsymbol{x} = \boldsymbol{y})$;
3. $\forall \boldsymbol{x}(\boldsymbol{x} + \mathbf{0} = \boldsymbol{x})$;
4. $\forall \boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x} + \boldsymbol{S}(\boldsymbol{y}) = \boldsymbol{S}(\boldsymbol{x} + \boldsymbol{y}))$;
5. $\forall \boldsymbol{x}(\boldsymbol{x} \cdot \mathbf{0} = \mathbf{0})$;
6. $\forall \boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x} \cdot \boldsymbol{S}(\boldsymbol{y}) = \boldsymbol{x} \cdot \boldsymbol{y} + \boldsymbol{x})$;
7. $\boldsymbol{\phi}(\mathbf{0}) \wedge \forall \boldsymbol{x}(\boldsymbol{\phi}(\boldsymbol{x}) \Rightarrow \boldsymbol{\phi}(\boldsymbol{S}(\boldsymbol{x}))) \Rightarrow \forall \boldsymbol{x} \boldsymbol{\phi}(\boldsymbol{x})$.

Our function $\boldsymbol{add^i}(\boldsymbol{A})$ is formalizable as bounded iteration of addition using the recursive rule for addition in PRA. Similarly, $\boldsymbol{mul^i}(\boldsymbol{A})$ uses power definition as repeated multiplication via induction.

3.3. Primitive Recursive Arithmetic (PRA) Considerations

Within Primitive Recursive Arithmetic, we define addition and multiplication using base cases and recursive rules:

$\boldsymbol{add}(\boldsymbol{x}, \boldsymbol{y})$via base case and successor iteration

$$\boldsymbol{add}(\boldsymbol{x}, \mathbf{0}) = \boldsymbol{x} \quad (8)$$

$$\boldsymbol{add}\{\boldsymbol{x}, \boldsymbol{S}(\boldsymbol{y})\} = \boldsymbol{S}\{\boldsymbol{add}(\boldsymbol{x}, \boldsymbol{y})\} \quad (9)$$

$\boldsymbol{mul}(\boldsymbol{x}, \boldsymbol{y})$via base case and successor iteration

$$\boldsymbol{mul}(\boldsymbol{x}, \mathbf{0}) = \boldsymbol{x} \quad (8)$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2025 doi:10.20944/preprints202507.2176.v1

$$mul\{x, S(y)\} = add\{x, mul(x, y)\} \tag{9}$$

Based on these definitions, our indexed unary system aligns as follows:

$$sub^i(A) = mul(A, 1 - i) \tag{10}$$

$$add^i(A) = mul(A, I + i) \tag{11}$$

$$mul^i(A) = A^{1+i} \tag{12}$$

$$div^i(A) = A^{1-i} \tag{13}$$

Each of these operations is defined recursively over the operand, and since they use only primitive recursion and composition, they are provably valid within PRA.

3.4. Clone-Theoretic Analysis

3.4.1. Closure

Composition of any two functions in the family results in another unary function with updated $\boldsymbol{i}$. Thus, closure under composition is satisfied [2]. A good example of closure shows the composition of $add^i(A)$ and $mul^j(A)$:

$$add^i\{add^j(A)\} = A((1 + i)(1 + j)) \in Ƒ \tag{14}$$

$$mul^i\{add^j(A)\} = \{A((1 + j)\}^{1+i} = (1 + j)^{1+i}A^{1+i} \tag{15}$$

3.4.2. Independence

Although the four arithmetic functions exhibit algebraic identities, none of the can be constructed by composing the others. We propose the following identities to be true for fixed $\boldsymbol{A}$ and $\boldsymbol{i}$:

$$add^i(A) + sub^i(A) = 2A \tag{16}$$

$$mul^i(A).\,div^i(A) = A^2 \tag{17}$$

$$\frac{add^i(A).\,sub^i(A)}{mul^i(A).\,div^i(A)} = 1 - i^2 \tag{18}$$

$$[div^i(A)]^{add^i(A)} = [mul^i(A)]^{sub^i(A)} \tag{19}$$

$$\frac{sub^i(A)}{add^i(A)} = \frac{1 - \mathrm{i}}{1 + \mathrm{i}} \tag{20}$$

Although algebraic relationships exist even among logically independent functions, these identities do not violate clone independence. It follows that:

- $add^i(A)$ grows linearly
- $mul^i(A)$ grows exponentially

This difference in growth classes ensures clone-theoretic independence [2,6].

3.5. Kalmar Correspondence

Our functions correctly reconstruct Kalmar's elementary primitives:

1. Zero: $add^{-1}(A) = 0$
2. Identity: $add^0(A) = A$

3. Successor Chain: $add^1(A) = 2A$
Bounded Sum

$$S_n(A) = \sum_{k=1}^{n} add^1(A) = 2An \tag{21}$$

Bounded Product

$$P_n(A) = \prod_{k=1}^{n} mul^1(A) = (A^2)^n = A^{2n} \tag{22}$$

Thus, Ƒ reconstructs Kalmar's bounded-sum and bounded-product class

3.6. *Lawvere and Gödelian Semantics*

3.6.1. Lawvere Theories

Based on category considerations, each function in Ƒ defines an endomorphism in a Lawvere theory [5]. The clone closure forms a subcategory where morphisms represent computable operations under composition. The minimality of Ƒ aligns well with categorical generators of arithmetic objects.

Unary operations $f: X \rightarrow X$ over $\mathbf{N}$ form endomorphisms. In our system $Ƒ \subseteq End(\mathbf{N})$.

- $add^i(A)$ is scaling by $1 + i$
- $mul^i(A)$ experiences exponentiation by $1 + i$

Under composition, F defines a subcategory, satisfying Lawvere axioms.

3.6.2. Gödelian Encoding

Gödel encodings allow each function $f \in Ƒ$ to be assigned a code $\#f$, producing a Gödel-indexed basis for PRA-compatible computation. The formal system, thus, has representability under both syntactic recursion and semantic functional mappings.

$$\#(add^i) = 2^{1+i} \tag{23}$$

$$\#(mul^i) = 3^{1+i} \tag{24}$$

Pairing:

$$(a, b) = 2^a 3^b \Rightarrow \#(f \circ g) = \pi(\#f, \#g) \tag{25}$$

Thus, functions and their composition are fully arithmetizable.

3.7. *Extension to Hyperoperations*

$$hyper_1{}^i(A) = add^i(A) \tag{26}$$

$$hyper_2{}^i(A) = mul^i(A) \tag{27}$$

$$hyper_3{}^i(A) = exp^i(A) \tag{28}$$

Recursively, all operations remain unary and maintain clone-theoretic properties; making the system suitable for embedding in Grzegorczyk or fast-growing hierarchies [6].

3.8. *Unary Minimality Theorem*

Theorem 1. *Let $Ƒ = \{add^i(A), sub^i(A), mul^i(A), div^i(A)\}$. Then Clone(Ƒ) is functionally complete for elementary arithmetic, and Ƒ is minimal under clone generation.*

Proof of Theorem 1. Proof

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 July 2025 doi:10.20944/preprints202507.2176.v1

1. Closure under composition:

If $\boldsymbol{f}, \boldsymbol{g} \in \mathfrak{F}$, then $\boldsymbol{f}\{\boldsymbol{g}(\boldsymbol{A})\}$ results in another scalar-parameterized function in $\mathfrak{F}$. Thus, $\mathbf{Clone}(\mathfrak{F})$ is closed.

2. Expressivity:

$$\boldsymbol{add}^{-1}(\boldsymbol{A}) = \mathbf{0} \tag{29}$$

$$\boldsymbol{add}^{0}(\boldsymbol{A}) = \boldsymbol{A} \qquad \text{provides identity} \tag{30}$$

$$\boldsymbol{add}^{0}(\boldsymbol{A}) = \boldsymbol{A} \ \text{ from which successor can be emulated using bounded offset} \tag{31}$$

3. Kalmar Functions Recovery:

$\boldsymbol{add}^{(1)}$ yields bounded sum. Similarly, $\boldsymbol{mul}^{(1)}$ yields bounded product. Hence, we recover Kalmar's base [1,3].

4. Incomposability:

Suppose, $\boldsymbol{mul}^{(i)}$ is reducible to $\boldsymbol{f}\{\boldsymbol{add}^{i}(\boldsymbol{A})\}$, then: $\boldsymbol{A}^{1+i} = \boldsymbol{f}\ (\boldsymbol{A}(\boldsymbol{j} + \mathbf{1}))$ but no unary function f can grow a linear input into an exponential output. This yields Contradiction. Similar arguments apply to the other unary pairs.

5. Minimality by Contradiction:

Suppose that $\mathfrak{F}' = \mathfrak{F}$, it would imply that $\boldsymbol{f}\{\boldsymbol{sub}^{i}(\boldsymbol{A})\}$ still generates $\mathbf{Clone}(\mathfrak{F})$. Hence, inverse-linear behavior is lost, and bounded difference becomes unrepresentable. Thus, F is minimal.

□

4. Discussion

This study postulates a novel unary functional system that addresses the Unary Function Clone Problem by defining indexed unary operations—each parameterized by an integer and closed under composition. These functionals, grounded in Primitive Recursive Arithmetic, reconstruct elementary arithmetic operations without requiring binary inputs or external constants. The results differ from traditional multivariate/ categorical methods by demonstrating a clone-independent minimal unary system exhibiting formal closure and expressibility. We adhere to Kalmar's class, Peano Arithmetic, Gödelian and Lawvere theories; instilling syntactic and semantic validity. Notably, the functional growth distinctions ensure clone independence, while algebraic relations between operations reveal internal structural consistency. This study aims to inspire further research into unary representations of complexity classes, functional encodings, and symbolic arithmetic frameworks.

Acknowledgments: During the preparation of this manuscript, for the purposes of contextual grounding and paraphrasing, the author utilized GenAI (ChatGPT). However, the author conceived all original mathematical constructs, theoretical creativity, innovations and interpretations. The author has reviewed and edited the output and takes full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

PRA	Primitive Recursive Arithmetic

Reference

1. Kalmar, L. (1959). An Argument Against the Plausibility of Church's Thesis.
2. Burris, S., & Sankappanavar, H.P. (1981). A Course in Universal Algebra.
3. Kleene, S.C. (1952). Introduction to Metamathematics.
4. Smoryński, C. (1991). Logical Number Theory I.

 doi:10.20944/preprints202507.2176.v1

5. Lawvere, F. (1963). Functorial Semantics of Algebraic Theories.
6. Rose, H.E. (1984). Subrecursion: Functions and Hierarchies.
7. M. Behrisch and H. Machida, "On Minimality of Some Binary Clones Related to Unary Functions," 2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL), Miyazaki, Japan, 2020, pp. 297–302. doi: 10.1109/ISMVL49045.2020.00062