Canonical Set Theory with Applications from Parallel Addition of Multiple Inputs to Matrix Multiplication and Data Structures

Juan Ramírez

Preprint v2

Aug 13, 2023

https://doi.org/10.32388/G6KRQE.2

Canonical Set Theory with Applications from Parallel Addition of Multiple Inputs to Matrix Multiplication and Data Structures

Juan Ramírez

Aug 13, 2023

Jalisco, México
www.binaryprojx.com
jramirez@binaryprojx.com

Abstract

Few questions that have resounded through the mind of the mathematician, as much as the simple question of describing the nature of a number. The longstanding consensus is that it does not matter which set theory is used to describe numbers. What matters is that it can be done. It is widely believed that the particular choice of a construction for natural and real numbers is irrelevant for the rest of mathematics. A set theory is proposed as a canonical theory that yields transparent proofs in fundamental areas of mathematics including group theory, discrete mathematics, analysis, data types, and new results tying these areas and addressing Hilbert's 24th (twenty-fourth) Problem and Benacerraf's Identification Problem. Applications to computer science are discussed and these include a linearly-scaleable circuit that serves for parallel addition and multiplication of scalars, vectors and matrices, with wide implications for Area-Specific Integrated Circuits (ASICs) used in CGI, Neural Networks and AI training, Digital Signal Processing, among other applications that depend on fast and low-powered vector operations. This low-power In-Situ (In-Memory) computing architecture, based on a patent-pending Simple and Linear Fast Adder (SLFA), is a direct consequence of the proposed set theory. Algebraic invariants are also described with results bringing together set theory, discrete mathematics, number theory and algebraic structures. A canonical block form is defined for the Cayley table of finite groups, in terms of a numeric representation of groups. Automorphisms, and the minimal independent system of equations that define the group are given by the block form, among other information regarding groups' internal and external structure. The proposed construction of natural numbers is generalized to provide a simple and transparent construction of the continuum of real numbers, with a fast approximation for the numeric derivative that can be implemented with the SLFA. Infinite data structures are defined in the most efficient way with the smallest possible data type. A countable sequence of real numbers is coded in a single real number, and an infinite $\infty \times \infty$ real-valued matrix is also coded with a single real number. A real function is coded in a set of real numbers, and a countable sequence of real functions is also coded in a set of real numbers. These codings are meaningful and computable. Mathematical objects of all types are well assigned to tree structures in a proposed hierarchy of types.

Keywords: Structuralism; Set Theory; Fast Adder; Arithmetic Logic Unit; Finite Group; Real Number; Fast Derivative; Data Types; Tree; Type Theory; Computability; Complexity, Matrix Multiplication

Introduction

The first section gives appropriate definitions for operation, group, field and linear space are given that allow simple constructions and proofs in the next sections. It is not a prerequisite for understanding the other sections. The reader who is not motivated by algebraic definitions may skip this section, but Definition 1 and Theorem 1, found in this section, should be understood before proceeding.

Today, every mathematician knows that the natural number zero is the empty set, $\emptyset$. However, most mathematicians in the time of Hilbert (and to this day) did not feel the need to justify the existence of the number 0, and argued that the nature of numbers is irrelevant and only their behavior is important. When Hilbert proposed his famous list of Problems (Paris, 1900), he paid special attention to the philosophical and practical implications of the Axiomatic Method; abundant material on this subject is found in [Corry(2010)] and other articles and books from this author. Hilbert gave so much thought to this type of problem that the first two and the sixth problem are axiomatic questions. The first is the Continuum Hypothesis, the second is the problem of proving consistency and completeness of arithmetic, and the sixth is the axiomatization of physics. In the decades that followed, a monumental, collective, attempt was made to answer these metamathematical questions through the formalization of different set theories and their logical structures. An explosion of new ideas, concepts and methods was born from these philosophical questions. Physics had been thought to be nearly explained away at the end of the 19-th century, but then relativity and quantum theory came along and pulled the veil. Then, just when everybody thought that a perfect description, if not of physics, but at least of mathematics, was possibly attainable, Gödel tore down these illusions with his incompleteness theorems. He did it in the preconceived world where the Peano Axioms are true. The incompleteness theorems are true if one assumes that natural numbers are objects ruled by Peano's axioms. It is a general consensus that the axiomatization of natural numbers chosen, or the specific computable coding of numbers and other structures, is irrelevant and all that matters is that it can be done. Still, the question has been raised in the literature of philosophy of mathematics, structuralism, and set theory, if an alternative axiomatic system could exist that solves, not necessarily all, but some of the metamathematical questions on the foundations of mathematics. Is there an alternative set theoretic definition of natural numbers that is specific enough that all mathematical objects are well defined but general enough that the widest possible collection of theorems is provable, all while carefully avoiding paradox? These types of questions have taken various forms, and it is difficult enough to ask them in any of these forms. One of the forms this question has taken is Benacerraf's Identification Problem [Benacerraf(1965)]. Hilbert had analyzed this problem in a similar way, but perhaps he thought it was not prudent to cause mathematicians more confusion than he was already going to cause with problems 1,2 and 6. Hilbert knew that simply asking these questions was enough of a complication to mathematics and that this next problem, his 24th problem [Thiele(2003)], could wait. *"The 24th problem in my Paris lecture was to be: Criteria of simplicity, or proof of the greatest simplicity of certain proofs. Develop a theory of the method of proof in mathematics in general. Under a given set of conditions there can be but one simplest proof."* A set theory is proposed that simplifies proofs and provides applications in pure and applied mathematics. The case is made that this is a unique and canonical set theory as part of a broader attempt in proposing an optimal universe for classical mathematics from number theory to analysis [Ramirez(2019)]. The second section describes natural numbers as the set of all hereditarily finite sets, **HFS**. An order $<$ and operations $\oplus, \odot$ are defined on **HFS**, isomorphic to the natural numbers $\mathbb{N}(<, +, \cdot)$. This axiomatization has important consequences in the representation and classification of finite and infinite objects.

A description for a "Simple and Linear Fast Adder" (SLFA-Patent Pending) sequential circuit with potentially comparable performance and more energy efficient than other fast adders [Uma(2012)],[Singh(2009)] is discussed in the second section. Appendix "A" is a self-contained patent description of the SLFA. The second section also describes a method for addition of multiple operands that reduces the addition of n-many operands to the addition of two operands, along with a description of the multiple input adder and its implementations in vector operations. This method is implemented in a simple design by connecting multiple SLFAs in parallel. Multiplication of two numbers is possible by using this architecture for addition of partial products, making it a good replacement of Carry Save Adders [Lutz(1994)] also. This multi-operation multi-operand circuit is achieved by connecting b-many SLFAs, of n-bits each, using only parallel connections. It is a rectangular grid of nodes that can perform addition of n-many b-bit numbers, or add b-many pairs of n-bit numbers, or multiply two numbers. The same circuit can also fast approximate numerical derivatives, creating more opportunities for fundamental computing advantages. One of the basic problems with Von Neumann Architecture is that memory and logic units are separate components with a relative distance between them and have to communicate back and forth through a bus. The bandwidth of the bus is usually the bottleneck of the processor. One solution proposed decades ago was the concept of Computing-In-Memory, but it has several implementation difficulties. The proposed rectangular ALU architecture offers solutions to some of the basic problems for Computing-In-Memory because the logic topology is identical to the memory topology (rectangular grid with parallel connections). Meaning, the architecture can be scaled in a one-to-one fashion with memory in terms of area, delay and topological complexity giving it an advantage in high-energy consuming tasks used in neural network training, processing SHA functions [Sun], etc. This architecture has significant advantages over traditional ALUs (Arithmetic Logic Units) in high-speed implementations of ASIC (Area-Specific Integrated Circuit) because the logic and memory have the same topology and can be layered one on top of the other in a one-to-one manner. Some of the advantages related to Computing-In-Memory are found in [Wang(2023)]. Specific designs are possible for scalar, vector and matrix multiplication pipelines using minimum area, delay and energy consumption than other CPU architectures [Hennessy(1990)]. The circuit design for vector and matrix operations will only be discussed very briefly here, given its potential to generate a competitive edge in the design of ASICs used in Digital Signal Processing, advanced CGI applications, Neural Networks and AI Training among other important applications that heavily rely on processing power for computing large numbers of matrix multiplications, faster and cheaper. Some details will be described here or in future publications which will be made available at the author's homepage, and other details will be reserved for collaboration. If you are interested in this line of research, or to become a financial partner, you can email or visit the author's homepage.

In the third section, a method for coding a finite function as a natural number is detailed. If A, B are two finite sets and $f : A \to B$ a function, then a unique natural number N_f is assigned to the function. A linear order on all finite functions is obtained that is well behaved in several ways. There is a suborder induced on the subset of all finite permutations which is also well behaved in its own ways. Specifically, if η_m, η_n are permutations of $m < n$ many objects, respectively, then $\eta_m < \mathbf{1}_n \leq \eta_n \leq \mathbf{id}_n$ where $\mathbf{1}_n$ is the one-cycle permutation of n objects and $\mathbf{id}_n$ is the identity permutation of n objects. This representation gives a good definition for equivalent functions and permutations. Two finite functions are equivalent if they are represented by the same natural number.

In the fourth section, a formal definition of finite groups is given in terms of natural numbers, where a single natural number is used to code the group in a computable manner. Every finite group G, is well represented with a natural number N_G; if $N_G = N_H$ then H, G are in the same isomorphism class. This defines a linear order on the set of all finite groups, that is well behaved with respect to cardinality. In fact, if H, G are two finite groups such that $|H| = m < n = |G|$, then $H < \mathbb{Z}_n \leq G$. The linear order on groups is

$$\mathbb{Z}_1 < \mathbb{Z}_2 < \mathbb{Z}_3 < \mathbb{Z}_4 < \mathbb{Z}_2^2 < \mathbb{Z}_5 < \mathbb{Z}_6 < D_6 < \mathbb{Z}_7 < \mathbb{Z}_8 < Q_8 < D_8 < \mathbb{Z}_2 \oplus \mathbb{Z}_4 < \mathbb{Z}_2^3 < \mathbb{Z}_9 < \mathbb{Z}_3^2 < \cdots , \qquad (1)$$

where D_n is the Dihedral group and Q_8 is the quaternion group. In general, $\mathbb{Z}_n \leq G$ if $|G| = n$ and the order is well behaved with respect to cardinality. The linear order induced on commutative groups, of n objects, also behaves well with respect to factorization of n. Intuitively, if $n = p^k$, then $\mathbb{Z}_{p^k} < \mathbb{Z}_p \oplus \mathbb{Z}_{p^{k-1}} < \mathbb{Z}_p^2 \oplus \mathbb{Z}_{p^{k-2}} < \cdots < \mathbb{Z}_p^k$. For example, $\mathbb{Z}_8 < \mathbb{Z}_2 \oplus \mathbb{Z}_4 < \mathbb{Z}_2^3$, and $\mathbb{Z}_9 < \mathbb{Z}_3^2$. If $n = p_1^{n_1} p_2^{n_2} p_3^{n_3} \cdots p_k^{n_k}$ is the prime factorization of n, then the commutative group $\mathbb{Z}_{p_1}^{n_1} \oplus \mathbb{Z}_{p_2}^{n_2} \oplus \mathbb{Z}_{p_3}^{n_3} \oplus \cdots \oplus \mathbb{Z}_{p_k}^{n_k}$ is the largest commutative group of n objects. For this purpose, a definition of canonical form for a group is given. The canonical form of a finite group is the Cayley table for the group, in a special block form. It reduces the problem of proving two finite groups are isomorphic to finding the canonical table of these groups. In the process of finding the canonical block form, the automorphisms and the minimal set of independent equations that define the group are obtained. An appendix is included where groups of less than ten objects are taken to their canonical block form. The canonical form and all twenty-four automorphisms of the symmetry group Δ_4 are also included in the appendix. A second appendix illustrates the canonical block form defined for finite groups.

The study of real numbers has been reduced to the study of natural numbers. However, the gap (conceptual and practical) between these two kinds of objects is enormous, in most treatments. The proposed set representation of natural numbers allows for the continuum of real numbers to be constructed as a natural extension of the set of natural numbers, without having to build intermediate structures such as $\mathbb{Z}$ or $\mathbb{Q}$. A natural number is a finite subset of **HFS**, while a real number is an infinite subset of **HFS**. A fast derivative algorithm is obtained as an approximation to the numerical derivative of a real function. Just as a finite group is reduced to a natural number, similar results are true in the infinite case. For example, a real function is a set of real numbers. More surprisingly, a countable sequence of real functions is also a set of real numbers. The general idea is that the complexity of objects is reduced to the minimum possible. In the last section, mathematical objects are well assigned to tree structures. Natural numbers are finite trees (objects of type 0), real numbers are infinite trees (objects of type 1). Sets of real numbers are objects of type 2, and a set of sets of real numbers is an object of type 4. A general description of types is briefly discussed.

Applications to be discussed in separate publications include a new coding of data structures in special memory units; an ALU architecture for optical computing schemes; a finite arithmetic important in the logical approach to artificial intelligence (AI) problems [Lovyagin(2021)]; a theoretical model on the probabilities of nuclear reactions, using a hexagonal grid that codes addition of natural numbers and multiplication of rationals through basic geometric relations; among others. The addition algorithm presented here relates the addition of numbers to the superposition of coupled waves; this is addressed in the conclusions. The lattice for determining nuclear reactions probabilities' with calculations on discrete number systems and the question of whether or not there exists an intrinsic connection between mathematics and physics, will be addressed in a later publication.

1 Groups, Fields and Linear Spaces

Operations are usually defined as a function of the form $(X \times X) \to X$. An alternate approach is taken here, by defining the operation of a group as a function $X \to (X \to X)$, which is known as Currying. A description of fields and linear spaces is also given in this section. The definitions and propositions, of this section, allow trivial proofs in the theory of set numbers of Section 2.

Definition 1. *Let G a non empty set, and $\boldsymbol{Aut}\, G$ the set of bijective functions of the form $G \to G$. A one-to-one function $G \to\ \boldsymbol{Aut}(G)$ is an operation on G. A set of functions $B \subseteq \boldsymbol{Aut}\, G$ is said to be balanced if $\mathbf{id}_G \in B$, and if $x \in B$ implies $x^{-1} \in B$. Let $* : G \to B$ a bijective function, for some balanced set $B \subset \boldsymbol{Aut}\, G$. If*

$$*(x) \circ *(y) = *(*(x)(y)), \tag{2}$$

for every $x, y \in G$, then $*$ is a group structure.

The functions $*(x)$ are called *operation functions of* $*$. The expression $*(x)(y) \in G$ is the image of y under the action of $*(x)$. Thus, $*(*(x)(y)) \in \mathbf{Aut}\ G$ is the image of $*(x)(y) \in G$ under the action of $*$.

Theorem 1. *The definitions of group and group structure are equivalent.*

Proof. Let $*$ a group structure and define an operation on the elements, $x * y = *(x)(y)$. It should be noted that $x*y = *(x)(y)$ is only a convention and depending on the specific function, this convention can vary. For example, an operation can be defined by $x * y = *(y)(x)$. The choice is irrelevant but must be consistent throughout, for each individual operation. Then, the following properties can be verified.

- Identity Element. There exists an object $e \in G$ such that $*(e) = \mathbf{id}_G$. Therefore, $*(e)(x) = x$ for all $x \in G$. This means $e * x = x$ for all $x \in G$. Now it must be shown $x * e = x$. It is true that $*(*(x)(e)) = *(x) \circ *(e) = *(x)$. Since $*$ is injective, it is also true that $*(x)(e) = x$.
- Inverse Element. Let $a \in G$, then there exists a unique $a^{-1} \in G$ such that $*(a^{-1}) = (*(a))^{-1}$ is the inverse function of $*(a)$. This is a direct consequence of the definition of balanced set. It will be proven that $a * a^{-1} = a^{-1} * a = e$. It is enough to prove $a^{-1} * a = e$. It can be verified that $a^{-1} * a = *(a^{-1})(a) = (*(a))^{-1}(a)$. Additionally, $*(a)(e) = a$. Therefore, the inverse function of $*(a)$ applies $(*(a))^{-1}(a) = e$.
- Associativity.

$$\begin{aligned} x * (y * z) &= *(x)(y * z) \\ &= *(x)(*(y)(z)) \\ &= (*(x) \circ *(y))(z) \\ &= *(*(x)(y))(z) \\ &= (*(x)(y)) * z \\ &= (x * y) * z. \end{aligned}$$

For the second part of this proof, it is enough to prove that a group G defines a group structure. The operation functions of the group structure are defined in terms of the cosets xG; define $*(x)$ by $g \mapsto_{*(x)} x * g$. It is easy to verify $*$ is an injective function and it is onto a balanced set. The associative property implies (2). □

The equivalence of groups and group structures is used to find their basic properties.

Theorem 2. *Let $G(*)$ a group with operation $*$. Then,*

1. *Right cancellation; $*(a)(c) = *(b)(c)$ implies $a = b$.*
2. *Left cancellation; $*(c)(a) = *(c)(b)$ implies $a = b$.*
3. *Uniqueness of identity and inverse elements.*
4. *Inverse of inverse; $(x^{-1})^{-1} = x$.*

5. *Existence of unique solutions; given $a, b \in G$ there exists a unique $x \in G$ such that $*(a)(x) = b$, and a unique $y \in G$ such that $*(y)(a) = b$.*

Proof. The first part requires to apply the function $*$, so that $*(*(a)(c)) = *(*(b)(c))$ which implies $*(a) \circ *(c) = *(b) \circ *(c)$. Right cancellation of functions gives $*(a) = *(b)$. It is concluded $a = b$ because $*$ is bijective. The second part can be proven similarly if left cancellation of functions is used.

Let e_1, e_2 be identity elements. Considering e_1 as identity, then $*(e_1)(e_2) = e_2$. If e_2 is the identity, then $*(e_1)(e_2) = e_1$. Therefore $e_1 = e_2$. The uniqueness of the inverse is trivial. If a_1, a_2 are inverse elements of a, then $*a(a_1) = e = *a(a_2)$ implies $a_1 = a_2$ because of left cancellation.

Let $y = x^{-1}$, so that $*(x)$ and $*(y)$ are inverse functions; $(*(x))^{-1} = *(y)$ and $(*(y))^{-1} = *(x)$. The inverse element of $y = x^{-1}$ is the object z such that $*(z)$ is the inverse function of $*(y)$. Therefore, x is the inverse of y and it is concluded $(x^{-1})^{-1} = x$.

For the last part, consider a, b fixed. Since $*(a)$ is a bijective function $G \to G$, there exists a unique $x \in G$ such that $*(a)(x) = b$. On the other hand, a function $*(y)$ that sends a to b needs to be defined. It is easy to see that $b * (a^{-1} * a) = b$, which can be rewritten as $(*(b) \circ *(a^{-1}))(a) = b$. The function $*(b * a^{-1}) = *(*(b)(a^{-1})) = *(b) \circ *(a^{-1})$ sends a to b so that $y = b * a^{-1}$ is the solution. Suppose there exists a second object, w, that satisfies the property of y. Then $*(y)(a) = *(w)(a)$ which implies $y = w$ if right cancellation is used. □

Proposition 1. *A group structure, $*$, defines a new function $\bar{*} : G \to \boldsymbol{Aut}(G)$ such that $\bar{*}(a)(b) = *(b)(a) = b*a$. The function $\bar{*}$ is also a group structure. The two group structures $*, \bar{*}$ are equivalent in the sense that they generate isomorphic groups.*

Proof. First prove $\bar{*}$ is a group structure. It must be shown $\bar{*}$ is a function $\bar{*} : G \to B$, where the image $Im\ \bar{*} = B$ is a balanced subset of $\mathbf{Aut}(G)$. Every object $a \in G$ is assigned a unique function $\bar{*}(a)$, and $\bar{*}(e) = \mathbf{id}_G$ for exactly one object $e \in G$. Next it will be proven $\bar{*}(a)$ is bijective. First of all, it is injective. Take $\bar{*}(a)(x) = \bar{*}(a)(y)$ which is equivalent to the expression $x*a = y*a$, then $x = y$ because of right cancellation. This proves $\bar{*}(a)$ is injective. To prove $\bar{*}(a)$ is onto G, let $b \in G$, then there exists a solution x to the equation $x * a = b$ which is equivalent to $\bar{*}(a)(x) = b$. This proves $\bar{*}(a)$ is a bijection. Now it will be proven the inverse function of $\bar{*}(a)$ is equal to $(\bar{*}(a))^{-1} = \bar{*}(a^{-1}) \in Im(\bar{*})$. By definition, $\bar{*}(a^{-1})(x) = x*a^{-1}$. Also, $\bar{*}(a)$ acts by $\bar{*}(a)(x*a^{-1}) = (x*a^{-1})*a = x$, which implies the inverse function $(\bar{*}(a))^{-1}$ acts by $(\bar{*}(a))^{-1}(x) = x*a^{-1}$. This proves $\bar{*}(a^{-1}) = (\bar{*}(a))^{-1}$. So far, it has been proven the image of $\bar{*}$ is a balanced set. To prove $\bar{*}$ is injective, take two objects $x, y \in G$ such that $\bar{*}(x) = \bar{*}(y)$. Then, $x = \bar{*}(x)(e) = \bar{*}(y)(e) = y$. Now show $\bar{*}$ satisfies the associative property. For all $a, b \in G$

$$\begin{aligned}
\bar{*}(\bar{*}(a)(b))(x) &= \bar{*}(b * a)(x) \\
&= x * (b * a) \\
&= (x * b) * a \\
&= \bar{*}(a)(x * b) \\
&= \bar{*}(a)(\bar{*}(b)(x)) \\
&= (\bar{*}(a) \circ \bar{*}(b))(x),
\end{aligned}$$

for all $x \in G$. This proves $\bar{*}$ is a group structure.

Let $G(*)$ be the group generated by $*$ and $G(\bar{*})$ the group generated by $\bar{*}$, then x^{-1} is the same inverse element under both operations. The inverse of $a * b$, under $*$, is equal to $b^{-1} * a^{-1}$. The inverse of $a * b = b \bar{*} a$, under $\bar{*}$, is equal to $a^{-1} \bar{*} b^{-1} = b^{-1} * a^{-1}$. These two groups are isomorphic by $x \mapsto x^{-1}$. To prove, take $\phi(a * b) = (a * b)^{-1} = b^{-1} * a^{-1} = \phi(b) * \phi(a) = \phi(a) \bar{*} \phi(b)$. □

Definition 2. *In general, the functions $*(x)$ and $\bar{*}(x)$ are not equal. When they are equal, the object x is said to commute. A group is abelian if its two generating functions are equal, $* = \bar{*}$.*

Proposition 2. *Let $G(*)$ an operation on the set G. The following are equivalent statements.*

1. *The operation $*$ is associative.*

2. *$*(*(x)(y)) = *(x) \circ *(y)$ for all $x, y \in G$.*

3. *$*(x) \circ \bar{*}(y) = \bar{*}(y) \circ *(x)$ for all $x, y \in G$.*

Proof. The equivalence of 1. and 2. was proven in Theorem 1. Prove the equivalence of 1. and 3. Let $z \in G$, then

$$\begin{aligned}
(*(x) \circ \bar{*}(y))(z) &= *(x)(\bar{*}(y)(z)) \\
&= *(x)(z * y) \\
&= x * (z * y) \\
&= (x * z) * y \\
&= \bar{*}(y)(x * z) \\
&= \bar{*}(y)(*(x)(z)) \\
&= (\bar{*}(y) \circ *(x))(z)
\end{aligned}$$

Suppose 3. holds, then associativity can be proven,

$$\begin{aligned}
x * (z * y) &= *(x)(z * y) \\
&= *(x)(\bar{*}(y)(z)) \\
&= (*(x) \circ \bar{*}(y))(z) \\
&= (\bar{*}(y) \circ *(x))(z) \\
&= \bar{*}(y)(*(x)(z)) \\
&= \bar{*}(y)(x * z) \\
&= (x * z) * y
\end{aligned}$$

□

The following result is useful for consequent sections. It gives a practical means of proving associativity. If the elements of G commute and the operation functions also commute, then the operation is associative.

Proposition 3. *If $*$ is a commutative operation on the set G, and $*(x) \circ *(y) = *(y) \circ *(x)$, for all $x, y \in G$, then $*$ is associative.*

Proof. Given the hypothesis, the equalities $*(x) \circ \bar{*}(y) = *(x) \circ *(y) = *(y) \circ *(x) = \bar{*}(y) \circ *(x)$ hold true. The result follows from 3. and 1. of the last proposition. □

Definition 3. *Let $G(*)$ a group and let $H \subseteq G$ be a subset of the set G. Define $*_H$ as the function $*$ restricted to H. If $*_H$ is a group structure then it is a subgroup of $G(*)$.*

For $H \subset G$ to be a subgroup of G it is necessary that the image of H, under the action of $*_H(h)$, be equal to H, for all $h \in H$. In short, $*_H(h)[H] = H$, for all $h \in H$. This means H *is closed under the operation* $*$.

Definition 4. *Given two groups $G_1(*_1)$ and $G_2(*_2)$, a homomorphism is a function $\phi : G_1(*_1) \to G_2(*_2)$ such that $\phi(*_1(a)(b)) = *_2(\phi(a))(\phi(b))$, for every $a, b \in G_1$. The set of all homomorphisms from $G_1(*_1)$ to $G_2(*_2)$ is represented by the notation $\boldsymbol{Hom}(G_1, G_2)$, when no confusion arises with respect to the operations of each group.*

If the homomorphism is injective as function then it is called a monomorphism, and if it is surjective as function it is called an epimorphism. If the function is bijective it is an isomorphism, or automorphism when $\phi : G \to G$. The set of all automorphisms of $G(*)$ is represented with the notation $\boldsymbol{Aut}\, G(*)$.

The notation $\mathbf{Aut}(G)$ and $\mathbf{Aut}\ G(*)$ is used to differentiate between bijective functions and automorphisms.

Theorem 3. *Let X a set, then the composition operation $\circ$ is a group structure for the set of all bijective functions $\boldsymbol{Aut}\ X$. A subset $B \subseteq \boldsymbol{Aut}\ X$ that is balanced and closed under composition is a subgroup $B(\circ) \subset \boldsymbol{Aut}\ X$.*

A group structure $* : G \to B$, induces an isomorphism $* : G(*) \to B(\circ)$.

The composition operation is a group structure for the set of automorphisms $\boldsymbol{Aut}\, G(*)$. A balanced and closed subset, $\mathcal{B} \subseteq \boldsymbol{Aut}\, G(*)$, is a subgroup $\mathcal{B}(\circ) \subset \boldsymbol{Aut}\, G(*)$.

Proof. For the first part, consider the function $\circ : \mathbf{Aut}\ X \to \mathbf{Aut}(\mathbf{Aut}\ X)$. If $f \in \mathbf{Aut}\ X$, then $\circ(f) : \mathbf{Aut}\ X \to \mathbf{Aut}\ X$ is the function that acts by $\circ(f)(g) = f \circ g$. It will be proven $\circ$ is a bijective function onto a balanced set $Im\ \circ$. Every object in $\mathbf{Aut}\ X$ is assigned a function $\circ(f) \in \mathbf{Aut}(\mathbf{Aut}\ X)$. To see $\circ$ is injective, take two objects $f, g \in \mathbf{Aut}\ X$ and suppose $\circ(f) = \circ(g)$. This implies $f = f \circ \mathbf{id}_X = g \circ \mathbf{id}_X = g$. Now, prove the image of $\circ$ is balanced. The identity of G is mapped to $\circ(\mathbf{id}_G) \in \mathbf{Aut}(\mathbf{Aut}\ X)$ which is the identity of $\mathbf{Aut}(\mathbf{Aut}\ X)$. Also, for every $\circ(f) \in \mathbf{Aut}(\mathbf{Aut}X)$, the inverse function is $(\circ(f))^{-1} = \circ(f^{-1}) \in \mathbf{Aut}(\mathbf{Aut}\ X)$. The associative property is the usual associativity of composition of functions. This proves the first assertion of the first part. The second assertion of the first part is trivial. Take $B(\circ)$ balanced and closed under composition. This makes $B(\circ)$ a group.

For the second part, it must be shown $*$ is an isomorphism. From the first part of this theorem, $B(\circ)$ is a group. It is also known $*$ is a bijection. Definition 4 and associativity, in G, are used to verify $*(*(x)(y)) = *(x) \circ *(y) = \circ(*(x))(*(y))$, for all $x, y \in G$. This proves that the group structure $*$ produces an isomorphism $G(*) \to B(\circ)$, where $B(\circ)$ is the image of $*$ with the operation $\circ$.

The third part of this theorem is proven similarly to the first part. □

The *distributive property* is defined. Rings and fields are also defined.

Definition 5. *Let $K(+)$ a group with identity 0; the set $K - \{0\}$ is represented by K_0. Let $\cdot : K_0 \to \mathcal{C} \subset$ **Hom**(K, K), an operation. The operation $\cdot$ distributes over $K(+)$, because*

$$\cdot(x)(+(a)(b)) = +(\cdot(x)(a))(\cdot(x)(b)),$$

for every $a, b, x \in K$.

Let $R(+)$ an abelian group, and let $\cdot$ a second operation that distributes over $R(+)$. Suppose $\cdot$ is associative and suppose $\cdot 1 = \boldsymbol{id}_R$ for a unique non trivial element $1 \in R_0$. A ring $R(+, \cdot)$ has two operations, and if $\cdot$ is commutative the ring is abelian.

*Let $K(+, \cdot)$ a ring and suppose $Im(\cdot) = \mathcal{C} \subset$ **Aut** $K(+)$ is a balanced set of automoprhisms. Then $K(+, \cdot)$ is a skew field. If the ring $K(+, \cdot)$ is abelian, $K(+, \cdot)$ is a field.*

A new notation $*x$ is used for the operation function $*(x)$. The distributive property holds when a group $K(\cdot)$ whose operation functions $\cdot x$, are homomorphisms on the original group $K(+)$. The conditions give the relations $\cdot x(0) = 0$, for all $x \in K$. Define $\cdot 0(x) = 0$. The operation function $\cdot 0$ is the trivial function $\mathbf{0} : K \to \{0\}$.

Corollary 1. *A field is an abelian group $K(+)$ together with a second abelian group $K(\cdot)$ that distributes over $K(+)$.*

Theorems 4 and 5, below, characterize linear spaces and modules. A linear space is an abelian group $V(\oplus)$, together with a field of automorphisms of $V(\oplus)$. Although these two theorems are not explicitly used in the following sections, it is useful for the last section on real numbers. Given an abelian group $V(\oplus)$ a second operation on $\mathbf{Hom}(V, V)$ is given, apart from composition. The operation $\oplus$ of V naturally induces a closed operation on $\mathbf{Hom}(V, V)$. This allows the definition of modules and linear spaces. Define addition of homomorphisms by $(f \oplus g)(x) = f(x) \oplus g(x)$. If $\mathcal{B} \subset \mathbf{Aut}\ V(\oplus)$, then the symbol $\mathcal{B}(\oplus)$ is used to emphasize that the set is being considered with addition, not composition. The trivial function $\mathbf{e} : V \to \{e\}$ acts as an identity object under addition of homomorphisms, $f = f \oplus \mathbf{e} = \mathbf{e} \oplus f$. Let $f \in \mathbf{Aut}\ V(\oplus)$, and $-f \in \mathbf{Aut}\ V(\oplus)$ the automorphism defined by $-f(x) = -(f(x))$ where $-(f(x))$ is the additive inverse of $f(x)$; the notation $-x$ is used for the inverse of x under $\oplus$. It is easily verified that $f \oplus (-f) = \mathbf{e}$. A set of automorphisms $\mathcal{B}(\oplus)$ is balanced if $\mathbf{e} \in \mathcal{B}(\oplus)$, and if $f \in \mathcal{B}(\oplus)$ implies $-f \in \mathcal{B}(\oplus)$.

Lemma 1. *Let $V(\oplus)$ an abelian group with identity e, and $\mathcal{B}(\oplus) \subset$ **Aut** $V(\oplus)$ a balanced set. If $\mathcal{B}(\oplus)$ is closed under addition of automorphisms, then $\mathcal{B}(\oplus)$ is an abelian group with identity $\boldsymbol{e}$.*

Proof. This result provides an easy way of knowing if $\mathcal{B}(\oplus)$ is a group with addition of functions. It is required that $\mathcal{B}(\oplus)$ be balanced. Under addition of automorphisms, the inverse of f is the function $-f$ that acts by $x \mapsto -(f(x))$. The inverse of $\mathbf{id}_V$ is $-\mathbf{id}_V$ that makes $x \mapsto -x$. Associativity in $V(\oplus)$ implies associativity in $\mathcal{B}(\oplus)$. The commutative property in $\mathcal{B}(\oplus)$ also follows from the commutative property in $V(\oplus)$. □

Theorem 4. *Let $V(\oplus)$ an abelian group and suppose $\mathcal{B}(\circ) \subset$ **Aut** $V(\oplus)$ is a balanced, closed and commutative set of automorphisms with composition. Suppose $\mathcal{B}(\oplus)$ is balanced and closed with addition. Then $\mathcal{B}(\oplus, \circ)$ is a field, and $V(\oplus)$ is a linear space over the field of automorphisms $\mathcal{B}$. The elements of $V(\oplus)$ are called vectors.*

Proof. With respect to composition, it is sufficient to verify $\mathcal{B}(\circ)$ is balanced, closed and abelian. From the third part of Theorem 3, it is concluded $\mathcal{B}(\circ)$ is an abelian subgroup of **Aut** $V(\oplus)$. If the conditions of the Lemma hold, then $\mathcal{B}(\oplus)$ is a group. Now it will be shown the distributive property holds. This is the simple statement that $\circ f$ is a homomorphism on $\mathcal{B}(\oplus)$, which is expressed by $f\circ(g\oplus h) = (f\circ g)\oplus(f\circ h)$ for every $f, g, h \in \mathcal{B}(\oplus, \circ)$. Let $x \in V$, then

$$\begin{aligned}(f\circ(g\oplus h))(x) &= f(g(x)\oplus h(x))\\ &= f(g(x))\oplus f(h(x))\\ &= (f\circ g)(x)\oplus(f\circ h)(x)\\ &= ((f\circ g)\oplus(f\circ h))(x).\end{aligned}$$

This proves $\mathcal{B}(\oplus, \circ)$ is a field. Now it will be proven the structure of a linear space, in the classic sense, has been defined. The scalar product is simply the application of an automorphism to a vector. Let $f \in \mathcal{B}$, then the scalar product of f, with a vector $v \in V$, is defined as $f \cdot v = f(v)$. First, $(f\circ g)(v) = f(g(v)) = f\cdot(g\cdot v)$ because $\circ$ is the product of the field. Also, $f\cdot(u\oplus v) = (f\cdot u)\oplus(f\cdot v)$ because $f \in$ **Aut** $V(\oplus)$. By definition of addition of functions, $(f\oplus g)\cdot v = (f\cdot v)\oplus(g\cdot v)$. A linear space is defined by an abelian group V and a set of automorphisms (of V) that form a field. □

Similarly define a module M over a ring.

Theorem 5. *Let $M(\oplus)$ an abelian group and suppose $\mathcal{B}(\circ) \subset \boldsymbol{Hom}(M, M)$ is a closed set of homomorphisms with composition, and $\boldsymbol{id}_M \in \mathcal{B}(\circ)$. Suppose $\mathcal{B}(\oplus)$ is balanced and closed. Then $\mathcal{B}(\oplus, \circ)$ is a ring. The group $M(\oplus)$ is a module over the ring of homomorphisms $\mathcal{B}$. In general, the group $\mathcal{B}(\circ)$ is not abelian.*

2 Finite Sets and Natural Numbers

Finding mathematical objects that satisfy the properties of order and operation for natural and real numbers is not an easy task. This problem was taken up by many mathematicians at the beginning of the last century to formalize arithmetic and analysis. The solution was found that the statements of arithmetic, and later analysis, can be formulated using an elementary concept, *set*. Attempts were made to find set representations of numbers and to model the structure of natural numbers, using sets. Being an elementary concept, a set is not described in terms of other mathematical objects. Rather, mathematical objects are described using the language of sets. A set is a special kind of *collection of objects*. However, in order to avoid paradoxes, the notions of naive set theory had to be formalized. A formal system consists of a formal language (alphabet and grammar), and a deductive system (logical axioms and inference rules). The alphabet is made up of letters $a, b, c, \ldots, A, B, C, \ldots$ used for representing the objects in question, and logical symbols necessary for the deductive system. In formalizing mathematics a logical approach is used, and the definition of formal system provides an abstract space for unintelligent/mechanized manipulation of symbols. A logical system is a formal system together with a set of non-logical axioms, and where the inference rules are first order logic or higher order logic. A classic example of a logical system is the Peano Arithmetic System. Even though the Peano Arithmetic System appeared to be sound at the beginning, many fatal flaws were pointed out in time. Extreme examples of these are Gödel's theorems. But, a simple and philosophical problem with the Peano axiomatization of natural numbers was that it did not provide answer to the question of what a number is. The entire edifice of mathematical objects is constructed based on the supposition of existence of 0, without saying exactly what object, if any, it is. The entirety of objects have their existence dependent on the existence of the object 0. If 0 exists then everything exists. That is why an even more foundational approach was given to the formalization of mathematics. The formal system of *Collections* which uses the letters $a, b, c, \ldots, A, B, C \ldots$ for collections. There is a single elementary binary relation. The symbol $\in$ is used for the binary relation of contention and the statement that a collection x is *element* of a collection X is represented with the symbol $x \in X$. Suppose the basic definitions for collections such as sub collection, arbitrary union of collections, arbitrary intersection of collections. A *Set Theory* is a logical system formed from the formal system of collections, plus non-logical axioms of set theory. Here, a set theory is proposed as a canonical basis for mathematics, in a non rigorous manner. It is similar to Zermelo-Fraenkel Set Theory but some of the set axioms are presented differently to construct natural numbers alternatively to Peano's Axioms, rendering simpler proofs, constructions and a range of practical applications.

The Zermelo-Fraenkel and Von Neumann constructions of natural numbers are the two most widely used definitions of natural numbers. In both cases, natural numbers are hereditarily finite sets and it can be proven that these sets satisfy Peano's Axioms. The set of all hereditarily finite sets, denoted **HFS**, consists of the sets obtained through the following procedure. The empty set is a member, $\emptyset \in \textbf{HFS}$. Also, if $x_1, x_2, \ldots, x_n$ are objects in **HFS** then $\{x_1, x_2, \ldots, x_n\} \in \textbf{HFS}$. Construct sets using these parameters to obtain all hereditarily finite sets. The collection $\{\emptyset\}$ is an object in **HFS**. Since $\emptyset$ and $\{\emptyset\}$ are in **HFS** the collection of these two objects, $\{\emptyset, \{\emptyset\}\}$, is also in **HFS**. Then, take $\emptyset$ and $\{\emptyset, \{\emptyset\}\}$ to find $\{\emptyset, \{\emptyset, \{\emptyset\}\}\} \in \textbf{HFS}$. The sets $\{\emptyset\}$ and $\{\emptyset, \{\emptyset\}\}$ help construct the set $\{\{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \in \textbf{HFS}$, etc. The first difficulty to formalize mathematics is to order hereditarily finite sets as natural numbers. The original methods are briefly discussed below, but it has been noted many times in the literature that these proofs and constructions of classic mathematical objects is quite long and cumbersome and involves artificial constructions. This leads to a lack of interest and basic comprehension from most mathematicians towards the axiomatic method and foundations of mathematics. Consequently, most mathematicians have a gap in the understanding of the nature of numbers, the basic blocks of mathematics. The mathematician probably knows that natural numbers can be built up in terms of finite sets, and that real numbers can be built using Cauchy Sequences of rationals, or Dedekind cuts, etc., but the conceptual understanding of these objects called numbers is difficult to grasp and widely considered to lack importance. Numbers are understood in terms of their interpretation as numbers, but hardly as objects in of themselves. The mathematical consensus is that the nature of a number is not important. It does not matter if we use one construction or another, what matters is that these constructions all describe objects whose properties and rules we can verify to satisfy the prerequisites of being a model of numbers. All that matters is that it can be done. It is known that any arithmetic axiomatization in first order logic is semantically incomplete but it is not generally true that the unprovable theorems for one axiomatization will be the same unprovable theorems for another axiomatization. One thing that is certain, however, is that different axiomatizations of numbers lead to different proof complexities for different theorems. In the two traditional constructions of natural numbers (Z-F and VN), discussed below, there are some advantages of one particular construction over the other in some aspects, and vice-versa. Here the argument is made that their exists a canonical set theory that yields new results and transparent proofs in many fundamental areas of mathematics including group theory, discrete mathematics, analysis, data types and computer science, among other areas. Definitions connecting these theories become apparent from the number construction proposed. A computable and meaningful function $\oplus 1 : \textbf{HFS} \to \textbf{HFS}$ that defines the order and operations of natural numbers will be the corner stone of our construction of natural numbers. Real numbers, and all other object types will be described in terms of natural numbers in a way that compuatble and meaningful representations and results of these objects can be given using the smallest possible data types.

The solution Zermelo and Fraenkel found is to order a sub collection of **HFS**. Notice it is trivial to order the sets $\emptyset$, $\{\emptyset\}$, $\{\{\emptyset\}\}$, $\{\{\{\emptyset\}\}\}, \ldots$, all of which are elements of **HFS**. If $x \in \textbf{HFS}$, then $\{x\} \in \textbf{HFS}$ is the successor. The order of natural numbers is trivially defined for $\mathbb{N}_< = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\{\{\emptyset\}\}\}, \ldots\}$. Addition of these sets has to be defined in such a way that it serves as a model of addition of natural numbers. This simply means the operation of addition has to be defined and its properties proven, which is usually tedious and laborious. But, the real difficulty arises in understanding the constructions and objects used to describe more complicated structures such as the integer numbers, rational numbers, and real numbers. Integers are described in terms of natural numbers. Rational numbers are described in terms of integers, and real numbers are defined in terms of rational numbers. The last step, in building real numbers, involves objects that are difficult to describe and work with. This leads to a gap in most undergraduate students' learning since most programs do not include these constructions. Even modern day efforts to describe the real number system do not provide an easy way to understand the nature of the object called *real number*.

A second approach in the formal description of natural numbers is due to Von Neumann. He also orders a subset of **HFS**. In particular, the sets $\emptyset < \{\emptyset\} < \{\emptyset, \{\emptyset\}\} < \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} < \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\} \ldots$. If x is a natural number, its successor $x + 1$ is the set $\{0, 1, 2, \ldots, x\}$. Here, $x < y$ if $x \in y$ which has some advantage in defining and proving properties of the order and addition. However, when building the later numerical structures, there is a similar situation as in the Zermelo-Fraenkel theory. The greater difficulty arises in building integers, rational numbers and real numbers. The constructions of Z-F and VN, and their technical aspects can be consulted in [Bernays(1991)].

The fact that there is at least two different constructions, gave way to another question, formally referred to as Benaceraff's Identification Problem. It has a great deal to do more with the Philosophy of Mathematics, than the mathematical models in use, but it still has wide implications. The main statement is set forth in a publication titled *"What Numbers Could Not Be"*, [Benacerraf(1965)]. The argument is made that numbers are actually not sets because there is no absolute way of describing them in terms of sets. For example, it cannot be known what object the number 3 is. Zermelo-Fraenkel say $3 = \{\{\{\emptyset\}\}\}$, but Von Neumann says $3 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$. Who is to be believed? In fact, there are infinitely many consistent set constructions of natural numbers. Are all these constructions on the same standing? Or are some more convenient than others? Both Z-F and VN provide injective functions $\mathbb{N} \to \mathbf{HFS}$. Ackermann was able to find a bijection $\mathbb{N} \to \mathbf{HFS}$. This is known as BIT-Predicate or Ackermann Coding [Ackermann(1937)], and it has practical implications since mathematical systems can be modeled directly in terms of classic computational processes. It also referred to as BIT-Predicate because it maps the natural number $\sum_i 2^{x_i}$ to the set $\{x_1, x_2, \ldots, x_n\}$. It is important to note that Ackermann coding itself does not give means for adding numbers in any special manner. Although Ackermann coding represents natural numbers as sets, it still treats numbers as binary sequences for purposes of addition and uses the traditional means of operating. Namely, carry over algorithms with its intrinsic time delays. This also makes it difficult to use Ackermann Coding as the basis for an axiomatic set theory. The reader is invited to attempt to find a simple description of the addition operation of natural numbers, in terms of elementary set operations. For example, the reader should try to define a computable and finite process that inputs A, B and outputs the set $A \oplus B$ which is the Ackermann coding for the sum of the numbers corresopnding to A and B? For eaxmple, if $A = \{\emptyset, \{\emptyset\}\} = 3$ and $B = \{\{\emptyset\}\} = 2$, the result of the process should be $5 = \{\emptyset, \{\{\emptyset\}\}\}$, and similarly for any two sets. Solutions to this are not trivial either; remember the goal is to express the operation in terms of elementary set operations (union and intersection of sets). This section is a proposal for a representation of natural numbers using BIT-Predicate, but addition is defined as a finite state machine that reaches stable state in logarithmic time, and the structure of natural numbers is defined by an alternate axiomatic base. The addition operation defined for sets can be easily extended from natural numbers to real numbers. A theory of types enveloping all classic mathematical objects is briefly discussed in the conclusions for later work.

2.1 Motivation

When adding two numbers, natural or real, there is one major difficulty involved. Addition is a special prefix problem which means that each sum bit is dependent on all equal or lower input bits, as noted in [Ladner and Fischer(1980)]. The carrying algorithm can also be consulted in [Metropolis, Rota and Tanny(1980)]. When adding numbers in base 10 (or base $b > 2$), sequences of digits must be used to represent natural numbers. To write a natural number in base b, each digit in the sequence will specify how many times the corresponding power of b is considered; digits will take a value in $\{0, 1, 2, \ldots, b-1\}$. The order of the sequence is important to know how many times each power of b is added. But, with binary representation ($b = 2$), a more elementary language suffices. It is no longer needed to specify how many times a power is added. It is sufficient to specify if a power is considered or not because digits of the sequence take values in $\{0, 1\}$. Essentially, this allows for a natural number to be determined by a *set* of smaller natural numbers; those that appear as power in binary form. For example, the number $7 = 2^0 + 2^1 + 2^2$ is the set $\{0, 1, 2\}$.

In this proposal, addition is treated in terms of sets, and not sequences. The sum $7 + 13 = (2^0 + 2^1 + 2^2) + (2^0 + 2^2 + 2^3)$, is the sum of sets $\{0, 1, 2\} \oplus \{0, 2, 3\}$. Two new sets are formed - symmetric difference and intersection. The powers that are not repeated $\{1, 3\}$, and the powers that repeat $\{0, 2\}$. To add a power of 2 with itself (i.e., numbers in the intersection), add "1" to that power, $2^n + 2^n = 2^{n+1}$. The sum is rewritten as $7 + 13 = (2^1 + 2^3) + (2^{0+1} + 2^{2+1})$. The first term $2^1 + 2^3$ represents symmetric difference $A \triangle B$, while the second term $2^{0+1} + 2^{2+1} = (2^0 + 2^2) + (2^0 + 2^2)$ represents the intersection. The sum has been reduced to $7 + 13 = (2^1 + 2^3) + (2^1 + 2^3)$. This step is iterated and the result is $7 + 13 = 2^{1+1} + 2^{3+1} = 2^2 + 2^4 = 20$. The system has reached a stable state because there are no more repeated powers, $\{0, 1, 2\} \oplus \{0, 2, 3\} = \{2, 4\}$.

This addition of finite sets is isomorphic to addition of natural numbers. To perform the addition of A, B form two new sets $A' = A \triangle B$ and $B' = s(A \cap B)$, where s is the function that adds 1 to the elements of its argument. The addition of these two new sets is the same as the original addition $A \oplus B = A' \oplus B'$ because it is equivalent to a rearrangement of the powers of 2. The terms A, B are rearranged into two new terms. The term A' consists of the non repeated powers (symmetric difference) and the term B' consists of the repeated powers (intersection). It is guaranteed that in a finite number of iterations the intersection $A^{(k)} \cap B^{(k)} = \emptyset$ becomes the empty set. This yields the final answer $A^{(k+1)}$, because $A \oplus B = A^{(k+1)} \oplus B^{(k+1)} = A^{(k+1)} \oplus s(\emptyset) = A^{(k+1)}$.

Apply this reasoning with another example, $15 + 23 = 38$, from Figure 1. This is the addition $A \oplus B =$

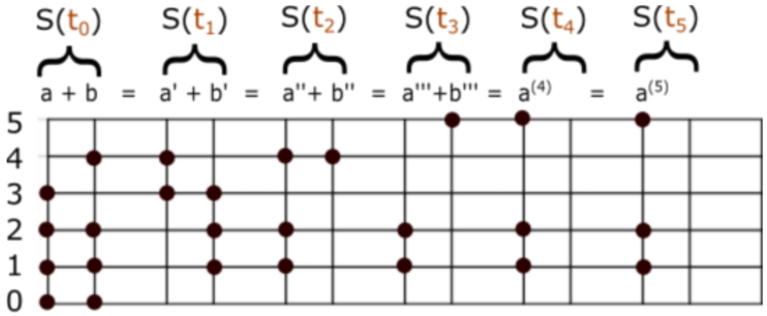

Figure 1: Graphic Representation of $15 + 23 = 38$. The sum of two sets is a process that ends in finite steps. The addition is iterated a finite number of times before the system stabilizes. In this example, the system stabilizes after three iterations. Observe that two disjoint set numbers form a stable system. This means $A \oplus B = A \cup B$ if $A \cap B = \emptyset$; the sum of disjoint sets coincides with the union.

$\{0, 1, 2, 3\} \oplus \{0, 1, 2, 4\}$ because $15 = 2^0 + 2^1 + 2^2 + 2^3$ and $23 = 2^0 + 2^1 + 2^2 + 2^4$. First find $A' = A \triangle B = \{3, 4\}$ and $A \cap B = \{0, 1, 2\}$, so that $B' = \{0 + 1, 1 + 1, 2 + 1\} = \{1, 2, 3\}$. Iterate the process with $A'' = A' \triangle B' = \{1, 2, 4\}$ and $B'' = s(A' \cap B') = \{3 + 1\} = \{4\}$. Continuing in this manner, a stable state is reached because $A''' \cap B''' = \{1, 2\} \cap \{5\} = \emptyset$.

The process described herein is a finite state machine. Each state is composed of two columns. Each column is a finite configuration of energy-levels representing one natural number, as is illustrated in Figure 1. A particle in the basic level "0" is worth 1 unit, and a particle in level "1" is worth 2 units. A particle in level "2" is worth 4 units, and in general a particle in level "n" is worth $2n$ units. A finite configuration of particles in a column represents a set number, so that each state is a pair of natural numbers. As shown in Figure 1, the initial state $S(t_0)$ is given by the inputs A, B. The next state, $S(t_1)$ is given by two new columns. The configuration of the left column is given by the energy levels that were not repeated in state $S(t_0)$. The right column in $S(t_1)$ is given by the objects that repeat but displaced one level up. The configuration of state $S(t_2)$ is defined similarly in terms of state $S(t_1)$. The left column of state $S(t_2)$ is given by the energy levels not repeated in state $S(t_1)$. The configuration in the right column of state $S(t_2)$ is given by the energy levels repeated in state $S(t_1)$ but displaced one level up. In general, the left column of state $S(t_{k+1})$ is given by the energy levels not repeated in state $S(t_k)$. The right column of state $S(t_{k+1})$ is given by a displacement, one level up, of the energy levels repeated in state $S(t_k)$. In a finite number of steps, a stable state is reached, where no particle occupies the right column. The result of the sum is given in the left column.

It should not be difficult for the reader to prove the number of steps to reach stability is bounded above by $\max(A \cup B) + 2$. The addition $A \oplus B = \{0, 1, 2\} \oplus \{0\}$ is one case that reaches the stable state in four steps (worst case scenario). Adding a unit to the string, $\{0, 1, 2, \ldots, k\} \oplus \{0\}$, gives the trivial result $\{k + 1\}$ in $k + 2$ steps. This is the set number expression for the equivalent arithmetical expression $1 + (1 + 2 + 4 + \ldots + 2^k) = 2^{k+1}$. In general, $\{n, n + 1, n + 2, \ldots, n + k\} \oplus \{n\} = \{n + k + 1\}$ is equivalent to the arithmetical expression $2^n + (2^n + 2^{n+1} + 2^{n+2} + \ldots + 2^{n+k}) = 2^{n+k+1}$. In fact, this type of string allows us to calculate the iterations for stability given a sum of two numbers. The longest string will give us the total number of iterations before stability. Going back to the bound on the number of iterations, it can be easily seen that it actually does not depend on the maximum value of the set. A more precise bound can be obtained. For example, the number of iterations for calculating $\{0, 1\} \oplus \{0\}$ is equal to the number of iterations for calculating $\{5, 6\} \oplus \{5\}$. However, the bounds are two very different numbers $\max\{0, 1\}$ and $\max\{5, 6\}$. To come up with a better bound on the number of iterations, observe that the number of iterations does not need to depend on how large the numbers are. To understand this, build a worst case scenario. Let A, B two sets such that $\#(A) + \#(B) = 3$. If the intersection $A \cap B = \emptyset$ is empty, the system is stable from the initial state. To maximize the number of iterations, build a string as above, $A = \{n, n+1\}$ and place the third element in the bottom $B = \{n\}$. This system requires a total of two iterations to stabilize. Any other configuration of three elements will require at most one iteration to stabilize. Now, suppose a total of $k+2$ objects; $\#(A)+\#(B) = k+2$. A string provides a worst case scenario; a string plus the smallest number of the string. The sum $A \oplus B = \{n, n+1, \ldots, n+k\} \oplus \{n\}$ takes a total of $k+2$ iterations to reach stability because of the string. Now, it is easy to see that there is more than one worst case scenario. Change one of the elements from A, to the set B, and the number of iterations will be the same. Doing this with $n+1$, the result is $\{n, n+2, n+3, \ldots, n+k\} \oplus \{n, n+1\}$ which will take $k+2$ iterations to stabilize. This can be done with any of the elements of A, and with more than one. In general, if $A \triangle B = \{n+1, \ldots, n+k\}$ and $A \cap B = \{n\}$, then exactly $k+2$ iterations are needed to stabilize. More generally, two sets with non empty intersection will have at least one string of this form. The longest of such strings will determine the smallest number of iterations needed for stability.

Suppose $A, B \subseteq \{0, 1, 2, \ldots, N-1\}$ are two random set numbers and let $x \leq N-1$. The probability that $x \in A \triangle B$ is equal to $\frac{2}{4} = \frac{1}{2}$. Then, the probability P that there exist $n, k \in \mathbb{N}$ such that $\{n+1, \ldots, n+k\} \subseteq A \triangle B$, is equal to the probability of k consecutive heads in N fair coin tosses. Therefore, the probability of a N-bit addition taking $k \leq N$ iterations to complete, is equal to $\frac{P}{4}$. On average, it takes $\log_2 N$ iterations to calculate a N-bit addition. The probability of taking more iterations than $\log_2 N$ decreases fast. These coin toss problems are standard. A Simple and Linear Fast Adder (Patent Pending) is described in the first appendix.

In the next subsection addition is formalized for finite sets, and it is isomorphic to addition of natural numbers $\mathbb{N}_+$. In the first section, a definition of operation was given that does not use a cartesian product in the domain. An operation is a function whose image is a space of functions itself. It is a one-to-one function $* : A \to (AfA)$ into the set AfA. The image, AfA, is the set of all one-to-one functions of the form $A \to A$. The operation $\oplus$ that defines addition of sets is defined in terms of its operation functions $\oplus n$ by $\oplus n(x) = n \oplus x$. The function $\oplus 1$ generates the hereditarily finite sets, and it also generates the set of operation functions $\oplus n$. The functions $\oplus n$ are the powers of composition, $\oplus 2 = \oplus 1 \circ \oplus 1$, $\oplus 3 = \oplus 1 \circ \oplus 1 \circ \oplus 1$, etc. Define two base cases $0 = \emptyset$ and $1 = \{\emptyset\}$, along with a function $\oplus 1 : \mathbf{HFS} \to \mathbf{HFS}$. To add 1 to a set A, apply the function $\oplus 1$ to the set A,

$$\oplus 1(A) = (A \triangle 1) \oplus s(A \cap 1), \tag{3}$$

where $s : \mathbf{HFS} \to \mathbf{HFS}$ sends every set $X = \{x\}_{x \in X}$ to the set $s(X) = \{\oplus 1(x)\}_{x \in X}$. Applying the function s to the set X simply means $\oplus 1$ is applied to every object of X. In the following calculations, use the fact that $s(\emptyset) = \emptyset$. Furthermore, define $A \oplus \emptyset = \emptyset \oplus A = A$ which simply defines $\emptyset$ as the identity element. First, use the definition of the operation to find $\oplus 1(0) = (0 \triangle 1) \oplus s(0 \cap 1) = 1 \oplus s(\emptyset) = 1 \oplus \emptyset = 1$. The function $\oplus 1$ generates every element of **HFS** when applied successively.

$$\begin{aligned}
2 &= \oplus 1(1) = (1 \triangle 1) \oplus s(1 \cap 1) = \emptyset \oplus s(1) = s(1) = \{\oplus 1(0)\} = \{1\} \\
3 &= \oplus 1(2) = (2 \triangle 1) \oplus s(2 \cap 1) = (\{1\} \triangle \{0\}) \oplus s(\{1\} \cap \{0\}) = \{0, 1\} \oplus s(\emptyset) \\
&= \{0, 1\} \oplus \emptyset = \{0, 1\} \\
4 &= \oplus 1(3) = (3 \triangle 1) \oplus s(3 \cap 1) = (\{0, 1\} \triangle \{0\}) \oplus s(\{0, 1\} \cap \{0\}) = \{1\} \oplus s(\{0\}) \\
&= \{1\} \oplus \{\oplus 1(0)\} = \{1\} \oplus \{1\}
\end{aligned}$$

A suitable definition for $\{1\} \oplus \{1\}$ must be found, and in general a suitable definition for $A \oplus B$ is needed. Extend the definition in the obvious way,

$$A \oplus B = (A \triangle B) \oplus s(A \cap B).$$

Now the number 4 can be found.

$$2 \oplus 2 = (2 \triangle 2) \oplus s(2 \cap 2) = \emptyset \oplus s(2) = s(2) = \{\oplus 1(1)\} = \{2\}.$$

This simply means the set $\{2\} = \{\{1\}\} = \{\{\{\emptyset\}\}\}$ is the object known as *the number 4*. Continue to generate sets, by applying the function $\oplus 1$ to the result.

$$\begin{aligned}
5 &= \oplus 1(4) = (4\triangle 1) \oplus s(4 \cap 1) = \{0,2\} \oplus s(\emptyset) = \{0,2\} \\
6 &= \oplus 1(5) = (5\triangle 1) \oplus s(5 \cap 1) = \{2\} \oplus s\{0\} = \{2\} \oplus \{1\} = (\{2\}\triangle\{1\}) \oplus s(\{2\} \cap \{1\}) \\
&= \{1,2\} \oplus s(\emptyset) = \{1,2\} \\
7 &= \oplus 1(6) = (6\triangle 1) \oplus s(6 \cap 1) = \{0,1,2\} \oplus s(\emptyset) = \{0,1,2\} \\
8 &= \oplus 1(7) = (7\triangle 1) \oplus s(7 \cap 1) = \{1,2\} \oplus s(\{0\}) = \{1,2\} \oplus \{1\} \\
&= (\{1,2\}\triangle\{1\}) \oplus s(\{1,2\} \cap \{1\}) = \{2\} \oplus s(\{1\}) = \{2\} \oplus \{2\} \\
&= (\{2\}\triangle\{2\}) \oplus s(\{2\} \cap \{2\}) = \emptyset \oplus s\{2\} = s(\{2\}) = \{3\} \\
9 &= \oplus 1(8) = (8\triangle 1) \oplus s(8 \cap 1) = \{0,3\} \oplus s(\emptyset) = \{0,3\} \\
10 &= \oplus 1(9) = (9\triangle 1) \oplus s(9 \cap 1) = \{3\} \oplus s(\{0\}) = \{3\} \oplus \{1\} = (\{3\}\triangle\{1\}) \oplus s(\{3\} \cap \{1\}) \\
&= \{1,3\} \oplus s(\emptyset) = \{1,3\}.
\end{aligned}$$

Notice, that the sum of two disjoint sets is the union. When referring to hereditarily finite sets, in this manner, they are called *set numbers*. Let N be a natural number with binary representation $\sum_{i=1}^{n} 2^{a_i}$, then N is the set number $\{a_1, a_2, \ldots, a_n\}$. For example, $5 = \{0,2\}$ because $5 = 2^0 + 2^2$, while $6 = \{1,2\}$ because $6 = 2^1 + 2^2$. The number $11 = \{0,1,3\}$ can easily be found.

$$11 \quad = \quad 5 \oplus 6 = \{0,2\} \oplus \{1,2\} = \{0,1\} \oplus s(\{2\}) = \{0,1\} \oplus \{3\} = \{0,1,3\}.$$

Another way of finding 11 is with the addition

$$11 = 7 \oplus 4 = \{0,1,2\} \oplus \{2\} = \{0,1\} \oplus s(\{2\}) = \{0,1,3\}.$$

2.2 Formalization

The constructions here described are carried out in a slightly modified version of Zermelo-Fraenkel Set Theory. The axioms needed for the constructions of this section are listed. The Axiom of Extensionality which defines equality of sets; two sets are equal if and only if they contain the same elements. The Axioms of Union and Subsets are also included; the Axiom of Subsets allows the construction of the intersection of sets.

To construct all hereditarily finite sets, from the empty set, their is a well defined procedure. Their exists a set $\mathbb{N}$ such that $\emptyset \in \mathbb{N}$, and if $x_1, x_2, \ldots, x_n$ are elements of $\mathbb{N}$ then $\{x_1, x_2, \ldots, x_n\} \in \mathbb{N}$. However, the objects of **HFS** are not generated in a particular order; there is no canonical order in constructing hereditarily finite sets. There are infinite ways of building these sets one by one. For example, once the sets $\emptyset$ and $\{\emptyset\}$ have been found, the sets $\{\{\emptyset\}\}$ and $\{\emptyset, \{\emptyset\}\}$ can be constructed. It is quite clear that 0 should be $\emptyset$ and 1 should be $\{\emptyset\}$. But, which of the two new sets should be the number 2? Ackermann Coding establishes that the number 2 is the set $\{\{\emptyset\}\}$, and 3 is the set $\{\emptyset, \{\emptyset\}\}$. Notice a fundamental difference the Ackermann Coding has with Z-F and VN constructions. Adding one unit to a Z-F or VN natural number is a simple procedure. In the first case, $\{x\}$ is the successor x, and in the second case the successor is $x \cup \{x\}$. With Ackermann Coding the situation is different because the rule for building new sets does not give an order to the sets built. Ackermann Coding builds sets without ordering them. To know the order of the hereditarily finite sets being built (the natural number corresponding to each element of **HFS**), the binary representation of natural numbers has to be known beforehand. The order given to finite sets is known only in hindsight when the binary representations of natural numbers is worked out. This was the difficulty in using Ackermann Coding as an axiomatic base of natural numbers. The problem with this was that there was no simple description of addition in terms of set operations. Numbers had to be operated as binary sequences which takes several layers of set theoretical constructions making the formalization long and difficult. The function $\oplus 1$ provided here is the first proposal found in the literature of a computable function that determines the Ackermann Coding successor of a hereditarily finite set, in terms of elementary set operations. A recursive set function $\oplus 1$ is proposed that defines addition of natural numbers in BIT-Predicate. This function depends on union and intersection of sets in **HFS** (the symmetric difference of sets can be expressed in terms of union and intersection). The set representation of every natural number is obtained by applying the function

$\oplus 1$ to $\emptyset$, a finite number of times. Moreover, the addition of two N-bit numbers is a finite state machine that reaches a stable state in $\log_2 N$ iterations, on average. For practical purposes in the implementation of addition in digital circuits, the Ackermann Coding of numbers is the most convenient and has therefore been referred to as BIT-Predicate. However, two natural numbers given in binary form are added with the carry-over algorithm which treats natural numbers as binary sequences, which introduces an intrinsic carry-over delay. These delays can be overcome with parallel computing algorithms but at a huge area and energy consumption cost, among other problems. An alternative method for parallel addition is given here that can be implemented with a simple and linear circuit with low-energy consumption; the patent-pending SLFA is found in the first appendix.

Definition 6. *Let $0 = \emptyset \in \boldsymbol{HFS}$ and $1 = \{\emptyset\} \in \boldsymbol{HFS}$. Define the set operation $m \oplus n$ with operation functions $\oplus n : \boldsymbol{HFS} \to \boldsymbol{HFS}$ such that $\oplus n(m) = m \oplus n = (m \triangle n) \oplus s(m \cap n)$, where $s(m \cap n) = \{\oplus 1(x)\}_{x \in m \cap n}$. In particular, the function $\oplus 1$ acts on sets by $\oplus 1(m) = (m \triangle \{\emptyset\}) \oplus s(m \cap \{\emptyset\})$.*

Let $n \in \boldsymbol{HFS}$ be the set obtained from $\oplus 1(\oplus 1(\cdots(\oplus 1(0))) = \oplus 1 \circ \oplus 1 \circ \cdots \circ \oplus 1(0)$. This can be expressed as $n = \oplus 1^n(0)$.

An axiom is also needed to ensure natural numbers are infinite. The infinity axiom proposed here is given in terms of a bijection $\oplus 1$. Let $\mathbb{N} = \mathbf{HFS}$, and $\mathbb{N}_1 = \mathbf{HFS}/\{\emptyset\}$ the set of hereditarily finite sets without the empty set. The Infinity Axiom is the statement that the function $\oplus 1 : \mathbb{N} \to \mathbb{N}_1$ is a bijection. This ensures the sets generated are infinite. The object 0 is sent to the object 1. Since 0 is not in the image set, it is not the image of 1. Also 1 cannot be the image of 1 because it is already the image of 0. This means a new object, call it 2, is the image of 1. The argument continues in this manner to prove there are infinitely many natural numbers. The set of arrows of $0 \to 1 \to 2 \to 3 \to \cdots$ are the ordered pairs of the function $\oplus$ that adds one unit, $x \to x \oplus 1$. If this set of arrows is extended to include transitive arrows, then the arrows give the total order of the natural numbers. The functions $\oplus 1^n$ and $\oplus n$ are both assigned to $n = \oplus 1^n(0)$. The equality $\oplus n = \oplus 1^n$ is taken as an axiom, for every $n \in \mathbb{N}$.

It is proven below that the operation functions $\oplus 1^n$ satisfy the properties of commutativity and associativity. To complete the operation of addition in terms of operation functions, the identity function is assigned to the empty set so that $\oplus 0(m) = m$ for every $m \in \mathbf{HFS}$. In the last sub section it has been illustrated how to find $\oplus 1(1), \oplus 1(2), \ldots$. When carrying out the calculations for $3 \oplus 1$, it was recognized that it is necessary to know the value of $\oplus 2(2)$. Continuing to apply $\oplus 1$, more calculations of the form $\oplus n(m)$ are encountered. But, the operation function for $\oplus n$ is explicitly dependent of $\oplus 1$. The functions $\oplus 1^n$ are defined as powers of $\oplus 1$, but to find $\oplus 1$ it is also needed to start finding $\oplus n$. The operation functions $\oplus 1^n$ and $\oplus n$ build each other simultaneously, as has been seen in the calculations of the previous section. The commutative property of $\oplus$ is trivial, using the fact that $f^n \circ f^m = f^m \circ f^n$ for a function f. Addition is $m \oplus n = \oplus n(m) = \oplus n(\oplus m(0)) = \oplus 1^n(\oplus 1^m(0)) = \oplus 1^m(\oplus 1^n(0)) = n \oplus m$. The reader may skip the proofs below, through Proposition 5, to the description of addition of several inputs and multiplication.

The easiest way to prove associative property of set addition is to prove the functions $\oplus m$ and $\bar{\oplus} n$ commute, for every set numbers m, n. Given that commutativity holds, it is true that $\oplus n = \bar{\oplus} n$. Because of Proposition 3, it is sufficient to prove the commutative property holds for operation functions, $\oplus m \circ \oplus n = \oplus n \circ \oplus m$.

Proposition 4. *The associative property holds for $\oplus$.*

Proof. By definition, the function $\oplus n$ is the function $\oplus 1$ applied a total of n times, $\oplus n(a) = \oplus 1^n(a)$. The operation functions $\oplus m$, $\oplus n$ commute,

$$\begin{aligned}(\oplus n \circ \oplus m)(a) &= \oplus n(\oplus m(a)) \\ &= \oplus 1^n(\oplus 1^m(a)) \\ &= \oplus 1^m(\oplus 1^n(a)) \\ &= \oplus m(\oplus n(a)) \\ &= (\oplus m \circ \oplus n)(a).\end{aligned}$$

□

A linear order has been given $0 \to_{\oplus 1} 1 \to_{\oplus 1} 2 \to_{\oplus 1} 3 \to_{\oplus 1} 4 \to_{\oplus 1} \cdots$, in terms of addition. The transitive arrows are aggregated to the set of arrows that defines the operation function $\oplus 1$. For example, since $0 \to 1$ is an arrow of $\oplus 1$, and $1 \to 2$ is an arrow of $\oplus 1$, then $0 \to 2$ is a transitive arrow. In the next section it will be specified exactly what is meant by an arrow or an ordered pair.

Let A, B two set numbers, then $A < B$ is true if and only if there exists a set number $m \neq \emptyset$ such that $B = A \oplus m$. Applying $\oplus n$ to B,

$$B \oplus n = \oplus n(A \oplus m) = \oplus n(\oplus m(A)) = \oplus m(\oplus n(A)) = \oplus m(A \oplus n) = (A \oplus n) \oplus m.$$

This implies $A \oplus n < B \oplus n$. That is to say, the operation preserves the order; $A < B$ implies $A \oplus n < B \oplus n$. The order is obviously transitive. Let $B = A \oplus m$ and $C = B \oplus n$. Then $C = (A \oplus m) \oplus n = A \oplus (m \oplus n)$. Since $m \oplus n$ is not the empty set, it is true that $A < C$.

The following result provides a practical way of determining the natural order of **HFS**. Let A, B two distinct natural numbers and consider their symmetric difference $A \triangle B$ which is not empty and is bounded. That is to say, $\max(A \triangle B)$ exists. Furthermore, this maximum is in exactly one of the two sets, not in both. Compare the two sets in terms of this object, $\max(A \triangle B)$. The set that contains this object is the largest of the two. For example, $15 = \{0, 1, 2, 3\} < \{4\} = 16$ because $A \triangle B = \{0, 1, 2, 3, 4\}$ and $\max(A \triangle B) \in \{4\} = 16$. The set number $A = \{1, 5, 6\} = 98$ is smaller than the set number $B = \{0, 7\} = 129$ because $\max(A \triangle B) = 7 \in B$. In the following proof it will be seen that the order is anti symmetric. It will also be seen that every pair of set numbers A, B is comparable; the order is total.

Theorem 6. *Let A, B two set numbers, then $A < B$ if and only if* $\max(A \triangle B) \in B$.

Proof. Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set number, and suppose B is a set number such that $A < B$. From (A5), the set number B is obtained by successively adding 1 to the set number A. This means $B = \oplus 1^n(A)$ for some $n \in \mathbb{N}$. It will be proven $\max(A \triangle B) \in B$ for every $B > A$. In this proof, the fact that $A \oplus B = A \cup B$, if $A \cap B = \emptyset$, will be used. Start with $A \oplus 1 = \{a_1, a_2, \ldots, a_n\} \oplus \{0\}$. There are two cases; $0 \notin A$ or $0 \in A$. In the first case, $A \oplus 1 = \{0, a_1, a_2, \ldots a_n\}$ which implies $\max(A \triangle (A \oplus 1)) = \max\{0\} = 0 \in A \oplus 1$. Now consider the second case; suppose $a_1 = 0$. Then, $A \oplus 1 = \{0, a_2, \ldots, a_n\} \oplus \{0\} = \{a_2, a_3, \ldots, a_n\} \oplus \{1\}$. There are two sub cases; $1 \notin A$ or $1 \in A$. In the first case, $A \oplus 1 = \{1, a_2, a_3, \ldots, a_n\}$ and the result follows, $\max(A \triangle (A \oplus 1)) = \max\{0, 1\} = 1 \in A \oplus 1$. In the second case, $a_2 = 1$ and this implies $A \oplus 1 = \{a_3, a_4, \ldots, a_n\} \oplus \{2\}$.

More generally, suppose k is the smallest number not in A. Then, $A = \{0, 1, \ldots, k-1, a_{k+1}, a_{k+2}, \ldots, a_n\}$, where $k < a_{k+1} < a_{k+2} < \ldots < a_n$. Applying $\oplus 1$ yields

$$A \oplus 1 = \{k, a_{k+1}, a_{k+2}, \ldots, a_n\}.$$

Then $\max(A \triangle (A \oplus 1)) = k$, which proves $\max(A \triangle (A \oplus 1)) \in A \oplus 1$. Applying $\oplus 1$ again, the result is

$$A \oplus 2 = \{k, a_{k+1}, a_{k+2}, \ldots, a_n\} \oplus \{0\} = \{0, k, a_{k+1}, a_{k+2}, \ldots, a_n\}.$$

This means $A \triangle (A \oplus 2) = \{1, 2, \ldots, k\}$ and the maximum is $k \in A \oplus 2$. Adding a unit again gives $A \oplus 3 = \{1, k, a_{k+1}, a_{k+2}, \ldots, a_n\}$, which then implies the symmetric difference is $A \triangle (A \oplus 3) = \{0, 2, 3, \ldots, k\}$ with maximum in $A \oplus 3$. Then, $A \oplus 4 = \{0, 1, k, a_{k+1}, a_{k+2}, \ldots, a_n\}$ and symmetric difference $A \triangle (A \oplus 4) = \{2, 3, 4, \ldots, k\}$. Continue in this manner, applying $\oplus 1$, until it has been applied a total of $2^k - 1$ times. In each step, the set A will be smaller than. Thus far, it has been proven $\max(A \triangle B) \in B$ if $A < B < A \oplus 2^k$. Applying $\oplus 1$ once more, is simply adding the singleton $2^k = \{k\}$ to the set A. The result is $A \oplus 2^k = \{0, 1, \ldots, k, a_{k+1}, a_{k+2}, \ldots, a_n\}$ because k is the smallest object not in A. This implies $\max(A \triangle (A \oplus 2^k)) = \max\{k\} = k \in A \oplus 2^k$. It is concluded $\max(A \triangle B) \in B$ if $A < B \leq A \oplus 2^k$. The careful reader will notice the elevator argument, or an induction hypothesis is needed to justify this argument. The rest is a repetition of what has been done up to this point. To apply $\oplus 1$ to $A \oplus 2^k$, simply substitute all the elements $0, 1, \ldots, k$ with $k+1$; use $2^{k+1} = 1 + (1 + 2 + 4 + 8 + \cdots + 2^k)$. There are two cases; $k+1 \notin A$ or $k+1 \in A$. In the first case, $\max(A \triangle (A \oplus 2^k \oplus 1)) = k+1 \in A \oplus 2^k \oplus 1$ because $A \oplus 2^k \oplus 1 = \{k+1, a_{k+1}, a_{k+2}, \ldots, a_n\}$. In the second case, $a_{k+1} = k+1$ so that

$$A \oplus 2^k \oplus 1 = \{k+1, a_{k+2}, \ldots, a_n\} \oplus \{k+1\}.$$

Proceed as before, finding the second smallest number not in A. Let $p \in A$ the smallest number in $A - \{k\}$. The numbers k, p are the two smallest numbers not in A, so that $A = \{0, 1, \ldots, k-1, k+1, k+2, \ldots, p-1, a_p, \ldots, a_n\}$. This implies $(A \oplus 2^k) \oplus 1 = \{p, a_p, \ldots, a_n\}$. The symmetric difference with A is $\{0, 1, \ldots, k-1, k+1, \ldots, p\}$. The maximum of the symmetric difference is $p \in (A \oplus 2^k) \oplus 1$. This proves $\max(A \triangle B) \in B$ if $A < B \leq (A \oplus 2^k) \oplus 1$. The symmetric difference of $(A \oplus 2^k \oplus 1) \oplus 1 = \{0, p, a_p, \ldots, a_n\}$ with A, is $\{1, 2, \ldots, k-1, k+1, k+2, \ldots, p-1, p\}$. The maximum of the symmetric difference is $p \in (A \oplus 2^k) \oplus 2$. This proves $\max(A \triangle B) \in B$, if $A < B \leq (A \oplus 2^k) \oplus 2$. Then, $(A \oplus 2^k) \oplus 3 = \{1, p, a_p, \ldots, a_n\}$, which again gives $p = \max(A \triangle (A \oplus 2^k \oplus 3)) \in A \oplus 2^k \oplus 3$. Continue in this manner. Apply $\oplus 1$ to $A \oplus 2^k$ a total of $2^k - 1$ times before reaching

$$(A \oplus 2^k) \oplus 2^k = \{0, 1, \ldots, k-1, p, a_p, \ldots, a_n\}.$$

Here, symmetric difference is $A \triangle ((A \oplus 2^k) \oplus 2^k) = \{k+1, k+2, \ldots, p\}$, and the maximum is $p \in (A \oplus 2^k) \oplus 2^k$. This proves $\max(A \triangle B) \in B$, if $A < B \leq A \oplus 2^k \oplus 2^k$. Adding 1 again, gives $A \oplus 2^k \oplus 2^k \oplus 1 = \{k, p, a_p, \ldots, a_n\}$. The symmetric difference with A is the set $\{k, p\}$. The maximum is $\max\{k, p\} = p \in A \oplus 2^k \oplus 2^k \oplus 1$. Keep adding 1 until $(A \oplus 2^k) \oplus 2^p = \{0, 1, \ldots, q-1, a_{n-q+1}, a_{n-q+2}, \ldots, a_n\}$ has been reached, where $A = \{0, 1, \ldots, k-1, k+1, \ldots, p-1, p+1, \ldots, q-1, a_{q-2}, a_{q-1}, \ldots, a_n\}$ and $q > p$ is the third smallest number not in A. This continues, for all $k, p, q, \ldots, r$ not in A. This proves $\max(A \triangle B) \in B$ if $A < B < A \oplus 2^k \oplus 2^p \cdots \oplus 2^r$. Upon adding 1 to $A \oplus 2^k \oplus 2^p \oplus \cdots \oplus 2^r = \{0, 1, \ldots, a_n\}$, the result is the singleton $\{a_n + 1\}$. It is trivial to prove that the maximum of the symmetric difference is in $A \oplus 2^k \oplus 2^p \cdots \oplus 2^r \oplus 1$, since $\max(A) = \{a_n\} < \{a_n + 1\} = \max(A \oplus 2^k \oplus 2^p \cdots \oplus 2^r \oplus 1)$. Observe that either $\max(X \oplus 1) = \max(X)$ or $\max(X \oplus 1) = \max(X) + 1$. Therefore, the result also holds for any $B > A \oplus 2^k \oplus 2^p \cdots \oplus 2^r \oplus 1$ because

$$\max(B) \geq \max(A \oplus 2^k \oplus 2^p \oplus \cdots \oplus 2^r \oplus 1) > \max(A)$$

.

To prove the second implication, use the following observation. Let $A = \{a_1, a_2, \ldots, a_n\}$ any set number, and let $b \notin A$ the maximum $b = \max(A \triangle B)$. Add 1 to the set number A repeatedly until you get to the set number $R = \{b, a_i1, a_i2, \ldots, a_n\}$, where $\{a_i1, a_i2, \ldots, a_n\}$ are the elements of A that are greater than b. This means a set N exists such that $R = A \oplus N$. Now, add P to R, where $P = \{b_1, b_2, \ldots, b_j\}$ is the set of objects in B that are smaller than b. The result is $B = P \oplus R = P \oplus (A \oplus N) = A \oplus (N \oplus P)$ which implies $A < B$. □

Let $A = \{2, 5, 6, 8, 9\}$ and $B = \{0, 1, 7, 8, 9\}$. The largest of the two is the set that contains $\max\{0, 1, 2, 5, 6, 7\} = 7$, so that $A < B$. In the next sections the order of set numbers will be given in a specific form. For example, a set number may be written in the form $A = \{\{3, 5\}, \{1, 2\}, \{4, 6\}\} = 2^{2^{2^3 + 2^5} + 2^{2^1 + 2^2} + 2^{2^4 + 2^6}}$. Compare it with $B = \{\{3, 4\}, \{1, 2\}, \{5, 6\}\} = 2^{2^{2^3 + 2^4} + 2^{2^1 + 2^2} + 2^{2^5 + 2^6}}$. The order relation is $A < B$ because $\max(A) = \{4, 6\} < \{5, 6\} = \max(B)$.

The operation function $\oplus 1$, of Definition 6, generates all **HFS** when applied successively to 0. The order in which sets are generated is an order of **HFS**, equivalent to the order of natural numbers $\mathbb{N}_{\leq}$. The operation function $\oplus n = \oplus 1^n$ is used to define addition of sets $\oplus 1^n(m) = m \oplus n = (m \triangle n) \oplus s(m \cap n)$.

2.3 Product of Set Numbers

The product is easy to define. Multiplication by 2 has already been defined. In binary representation $2^n + 2^n = 2^{n+1}$, and set numbers have a corresponding rule. To multiply by 2 is to apply the function $\odot 2 = s$ that adds 1 to the elements of the argument. Multiplication by 4 is $s \circ s$ which adds 2 to the elements of the argument. In general, multiplication of B by 2^k is equal to $s^k(B)$. If $B = \{b_1, \ldots, b_n\}$ then $2^k \odot B$ is equal to the set $\{b \oplus k\}_{b \in B} = \{b_1 \oplus k, \ldots, b_n \oplus k\}$. The product of a set number B with 2^k, in our graphic representation, consists of displacing the objects of the set, k units up. The set number $2^k \odot B$ is the *k-displacement of B*. The general product $A \odot B$ is defined in terms of displacements of the *base* B, and the *pivot* A.

$$A \odot B = \bigoplus_{a \in A} \{b \oplus a\}_{b \in B}. \tag{4}$$

Displacements of B are added, one for each object of the pivot A. If $a \in A$ then the a-displacement of B is one of the displacements in our sum. Notice that multiplication by 0 results in the empty set, $0 \odot X = X \odot 0 = 0$.

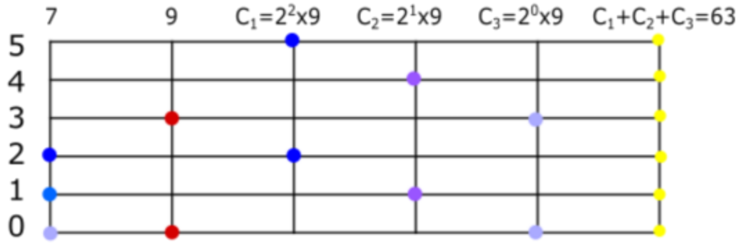

Figure 2: The product $7 \odot 9$. The first and second columns are the pivot and base, respectively. The next three columns correspond to the displacements of the base. The last column is the sum of the displacements. The result is equal to $63 = \{0, 1, 2, 3, 4, 5\}$.

It is also trivial to find $1 \odot X = X \odot 1 = X$. To show that $2 = \{1\}$ is commutative under multiplication,

$$\begin{aligned}\{1\} \odot X &= \{x \oplus 1\}_{x \in X} \\ &= \bigcup_{x \in X} \{x \oplus 1\} \\ &= \bigoplus_{x \in X} \{1 \oplus x\} \\ &= X \odot \{1\}.\end{aligned}$$

This means $2 \odot X = X \odot 2 = X \oplus X$. To find the product $7 \cdot 5 = (2^0 + 2^1 + 2^2)(2^0 + 2^2)$ use distribution to obtain $2^0(2^0 + 2^2) + 2^1(2^0 + 2^2) + 2^2(2^0 + 2^2)$. Then, $(2^{0+0} + 2^{2+0}) + (2^{0+1} + 2^{2+1}) + (2^{0+2} + 2^{2+2}) = (2^0 + 2^2) + (2^1 + 2^3) + (2^2 + 2^4)$. Carrying out the addition gives $7 \cdot 5 = 2^0 + 2^1 + 2^2 + 2^5 = 35$. Before proving general properties, calculate $5 \odot 15 = \{0, 2\} \odot \{0, 1, 2, 3\}$ in two different ways to verify these numbers commute. First make $A = 5$ and $B = 15$. Two displacements of $B = \{0, 1, 2, 3\}$ will be added. The first displacement is $\{0 \oplus 0, 1 \oplus 0, 2 \oplus 0, 3 \oplus 0\} = \{0, 1, 2, 3\}$, and the second displacement is $\{0 \oplus 2, 1 \oplus 2, 2 \oplus 2, 3 \oplus 2\} = \{2, 3, 4, 5\}$. Adding the two, gives $\{0, 1, 2, 3\} \oplus \{2, 3, 4, 5\} = \{0, 1, 3, 6\} = 75$. Now, make $A = 15$ and $B = 5$. Four displacements of $5 = \{0, 2\}$, each corresponding to an element of $15 = \{0, 1, 2, 3\}$. The displacements of 5 are $\{0, 2\}, \{1, 3\}, \{2, 4\}, \{3, 5\}$. Adding these four displacements results in $(\{0, 2\} \oplus \{1, 3\}) \oplus (\{2, 4\} \oplus \{3, 5\}) = \{0, 1, 2, 3\} \oplus \{2, 3, 4, 5\} = 75$, using associativity of addition.

Figure 2 shows the graphic representation of $7 \odot 9$. To formalize this, first verify $\odot 2$ is a morphism for addition of set numbers; verify $s(A \oplus B) = s(A) \oplus s(B)$. Use $X \oplus X = s(X)$ to prove $s(A \oplus B) = (A \oplus B) \oplus (A \oplus B) = (A \oplus A) \oplus (B \oplus B) = s(A) \oplus s(B)$. This implies

$$s^k(A \oplus B) = s^k(A) \oplus s^k(B), \tag{5}$$

for every $k \in \mathbb{N}$. To prove the distributive property use (5) and the commutative and associative properties of addition of sets.

$$\begin{aligned}A \odot (B \oplus C) &= \bigoplus_{a \in A} \{x \oplus a\}_{x \in B \oplus C} \\ &= \bigoplus_{a \in A} s^a(B \oplus C) \\ &= \bigoplus_{a \in A} (s^a(B) \oplus s^a(C)) \\ &= \bigoplus_{a \in A} s^a(B) \oplus \bigoplus_{a \in A} s^a(C) \\ &= (A \odot B) \oplus (A \odot C)\end{aligned}$$

To prove multiplication is commutative, let $a \in A$ fixed. The set $\{b \oplus a\}_{b \in B} = \{b_1 \oplus a, b_2 \oplus a, \ldots, b_n \oplus a\} = \{b_1 \oplus a\} \oplus \{b_2 \oplus a\} \oplus \ldots \oplus \{b_n \oplus a\}$ can be expressed as a sum of disjoint singletons, $\bigoplus_{b \in B} \{b \oplus a\}$. Therefore,

$$
\begin{aligned}
A \odot B &= \bigoplus_{a\in A} \{b \oplus a\}_{b\in B} \\
&= \bigoplus_{a\in A} \bigoplus_{b\in B} \{b \oplus a\} \\
&= \bigoplus_{b\in B} \bigoplus_{a\in A} \{a \oplus b\} \\
&= \bigoplus_{b\in B} \{a \oplus b\}_{a\in A} \\
&= B \odot A.
\end{aligned}
$$

The commutative property of addition and multiplication of sets has been proven. Together with the distributive property, these imply

$$(A \oplus B) \odot C = (A \odot C) \oplus (B \odot C). \tag{6}$$

Now it can be proven that the associative property holds for the product of set numbers. Because of Proposition 3, it is sufficient to verify the operation functions of $\odot$ commute. This can easily be done using mathematical induction. It is trivial to verify $\odot A$ and $\odot B$ commute for $N = 1$; it follows from the commutative property $\odot A \odot B(1) = A \odot B = b \odot A = \odot B \odot A(1)$. Suppose $\odot A$ and $\odot B$ commute for arbitrary N, so that $A \odot (B \odot N) = B \odot (A \odot N)$.

$$
\begin{aligned}
(\odot A \circ \odot B)(N \oplus 1) &= A \odot (B \odot (N \oplus 1)) \\
&= A \odot (B \odot N \oplus B \odot 1) \\
&= A \odot (B \odot N) \oplus A \odot B \\
&= B \odot (A \odot N) \oplus B \odot A \\
&= B \odot (A \odot N \oplus A \odot 1) \\
&= B \odot (A \odot (N \oplus 1)) \\
&= (\odot B \circ \odot A)(N \oplus 1)
\end{aligned}
$$

This proves associativity of multiplication. The next result characterizes multiplication as a repeated addition.

Proposition 5. *The operation function $\odot N$ acts on sets by $\odot N(X) = \oplus X^N(0)$.*

Proof. This is proven by mathematical induction on N. It is true for $N = 1$, since $1 \odot X = X$. Suppose it is true for N, then using the distributive and commutative properties

$$
\begin{aligned}
\odot(N \oplus 1)(X) &= \odot N(X) \oplus \odot 1(X) \\
&= \oplus X^N(0) \oplus X \\
&= \oplus X(\oplus X^N(0)) \\
&= \oplus X^{N+1}(0).
\end{aligned}
$$

□

It has been seen that multiplication is equivalent to the addition of multiple displacements of a given set number, or as the repeated addition of the same number. In ether case, it is the addition of multiple operands. The sub unit responsible for adding the partial products is usually referred to as accumulator of partial products. Some of the current proposals for multiplication are found in [Abrar(2019)], [Emmart(2011)], and [Taib(2020)]. A general method for defining the sum of multiple operands is proposed that has several advantages in hardware implementation. An algorithm is described that reduces the sum of 2^k summands to the sum of $k+1$ summands. In general, it reduces the sum of n summands to $\max(n) + 1$ summands. Consider the sum of 4-many, 8-bit numbers. The summands are $A = a_0a_1 \cdots a_7$, $B = b_0b_1 \cdots b_7$, $C = c_0c_1 \cdots c_7$, $D = d_0d_1 \cdots d_7$. This can be represented by the array

$$\begin{array}{cccc}
a_7 & b_7 & c_7 & d_7 \\
a_6 & b_6 & c_6 & d_6 \\
a_5 & b_5 & c_5 & d_5 \\
a_4 & b_4 & c_4 & d_4 \\
a_3 & b_3 & c_3 & d_3 \\
a_2 & b_2 & c_2 & d_2 \\
a_1 & b_1 & c_1 & d_1 \\
a_0 & b_0 & c_0 & d_0,
\end{array}$$

where each a_i, b_i, c_i, d_i is either 0 or 1. Count the number of 1's in each row. It takes three bits to write in binary form the number of 1's in a single row because $\max(4) + 1 = 2 + 1 = 3$. The number of 1's in each row can be represented in a 8×3 grid,

$$\begin{array}{ccc}
a'_7 & b'_7 & c'_7 \\
a'_6 & b'_6 & c'_6 \\
a'_5 & b'_5 & c'_5 \\
a'_4 & b'_4 & c'_4 \\
a'_3 & b'_3 & c'_3 \\
a'_2 & b'_2 & c'_2 \\
a'_1 & b'_1 & 0 \\
a'_0 & 0 & 0
\end{array} \tag{7}$$

The elements a'_0, b'_1, c'_2 will be used to write the number of 1's in row 0. The elements a'_1, b'_2, c'_3 are used to write the number of 1's in row 1, and elements a'_2, b'_3, c'_4 are used to write the number of 1's in row 2, etc. This maintains the representation of energy-levels and their unit value, while avoiding any intervention with totals from one row and the another. The three column grid can be reduced to two columns, by iterating the process. The total number of units in a single row of (7) will be represented with two bits because $\max(3)+1 = 1+1 = 2$. Addition of the two columns

$$\begin{array}{cc}
a''_7 & b''_7 \\
a''_6 & b''_6 \\
a''_5 & b''_5 \\
a''_4 & b''_4 \\
a''_3 & b''_3 \\
a''_2 & b''_2 \\
a''_1 & b''_1 \\
a''_0 & 0
\end{array}$$

is equivalent to the original four-input addition. Elements a''_0 and b''_1 represent the total value of the first row in (7). Elements a''_1 and b''_2 represent the total value of the second row, elements a''_2 and b''_3 represent the total value of the third row, etc.

An example is provided, to find the total of $A = 63 = \{0, 1, 2, 3, 4, 5\}, B = 37 = \{0, 2, 5\}, C = 21 = \{0, 2, 4\}, D = 38 = \{1, 2, 5\}, E = 28 = \{2, 3, 4\}, F = 13 = \{0, 2, 3\}, G = 14 = \{1, 2, 3\}, H = 52 = \{2, 4, 5\}$. This is given by

$$\begin{array}{cccccccc}
1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 & 0.
\end{array}$$

Since there can be at most eight objects in each row, only four bits are needed per row. This means the new grid has four columns. There is a total of $4 = \{2\}$ many number 1's in row 0. This is represented by placing the sequence of digits 0010 in the bottom most diagonal, of the new grid.

$$\begin{array}{cccc}
 & & & 0 \\
 & & 1 & 0 \\
 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.
\end{array}$$

Next, there is a total of $3 = \{0, 1\}$ number 1's in row 1. This is represented by placing the sequence of digits 1100 in the next diagonal.

$$\begin{array}{cccc}
 & & & 0 \\
 & & 0 & 0 \\
 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.
\end{array}$$

Since there are $8 = \{3\}$ number 1's in row 2, the sequence 0001 is placed in the next diagonal.

$$\begin{array}{cccc}
 & & & 1 \\
 & & 0 & 0 \\
 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.
\end{array}$$

Rows 3,4, and 5 have $4 = \{2\}$ number 1's each so that the sequence 0010 is placed in each of the following diagonals.

$$\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.
\end{array}$$

The sum of these four columns can now be reduced to the sum of three columns because three bits are enough for representing a total of four objects per row. Row 0, of the last grid, has a total of 0 number 1's so that the sequence 000 is placed on the bottom diagonal.

$$\begin{array}{ccc}
 & & 0 \\
 & 0 & 0 \\
0 & 0 & 0.
\end{array}$$

There is a total of $1 = \{0\}$ number 1's in Row 1 so that the sequence 100 is placed on the next diagonal.

$$\begin{array}{ccc}
 & & 0 \\
 & 0 & 0 \\
1 & 0 & 0 \\
0 & 0 & 0.
\end{array}$$

Continuing in this manner gives

$$\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
1 & 0 & 0 \\
1 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
1 & 0 & 0 \\
0 & 0 & 0.
\end{array}$$

The addition of three columns is reduced to $\max(3) + 1 = 2$ columns,

$$\begin{array}{cc}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
1 & 1 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 0.
\end{array}$$

Applying addition of two columns gives $A \oplus B \oplus \cdots \oplus H = 266$,

$$\begin{array}{cc}
0 & 0 \\
0 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 0.
\end{array}$$

The sum of n-many b-bit numbers can be computed with nb many nodes organized in a rectangular grid of size $n \times b$. This grid can be used to add n-many b-bit numbers, and the multiplication algorithm can also be executed. Parallel connections (only nodes from the same row or column are connected) allow for b-many n-bit SLFAs to perform addition of n-many b-bit numbers. The same low-powered circuit also performs parallel-multiplication of two inputs. This modified SLFA, its connections, extensions for vector and matrix multiplication along with case by case analysis and implementation proposals will be described in a separate paper, exclusively found in the author's personal web page "www.binaryprojx.com". To add n-many b-bit numbers, b-many n-bit SLFAs are placed side-by-side in a rectangular grid. Each node has two bits of memory belonging to the SLFA, and requires a third bit of memory that will be referred to as the principal bit. This means each node will consist of a three bit register and a Half Adder, maintaining the gate count and depth very low. The rectangular array requires parallel connections between nodes of the same row or column. The principal bits of the nodes are used for storing the initial inputs. Each SLFA will count the number of 1's in its row; one by one, send signals of the principal bits of that row to the SLFA. This is done in parallel so that all rows are counted at once and they can each signal process termination individually. Once the elements of a row are counted, the results stored in the SLFA will be sent to their new principal bits. In each iteration the columns with non-zero entries in their principal bits are less. This process is iterated until only two columns are non-zero in the principal bits. Given that this circuit performs addition of multiple inputs, it is also capable of carrying out multiplication of two binary inputs. Additionally, the circuit is able to perform parallel addition of b-many pairs of n-bit numbers, because each SLFA (rows) can be used as an independently-timed SLFA. The circuit is linearly scaleable in terms of bits and inputs, it presents the minimum possible topological complexity (rectangular grid of nodes with parallel connections), and is low-powered due to the gate depth. First, the number of 1's in each row is counted. This is done by sending signals from the principal bits into the the SLFA. When the i-th significant bit of a row is sent to the SLFA, the addition that follows takes at most $\max(i) + 1$ iterations of the SLFA, because the set number i requires $\max(i) + 1$ many digits to express in binary form. The worst case scenario, when adding n inputs, occurs if a row has n-many $1's$ in the principal bits. The number of steps in the worst case scenario is bounded by $\max(2) + \max(3) + \max(4) + \ldots + \max(n) + n$, but is much lower because most of the terms can be bounded by smaller numbers. For example, if i is a multiple of 2, then only one iteration of the SLFA is needed in that step. The only occasions where $\max(i) + 1$ iterations of the SLFA are needed is if i is of the form $2^k - 1$, for some k. Although PASTA adders [Rahman(2013)] are topologically equivalent to the SLFA, it is important to note the PASTA adder is an asynchronous circuit, like most fast-adders [Franklin(1994)], and therefore it lacks the memory units necessary for this addition of multiple inputs. The PASTA adder has multiplexers instead of the registers used in the SLFA, making it inappropriate for implementation in this multi-operand arithmetic architecture.

The relative efficiency of this implementation with respect to other circuits could be done by cases, in terms of the quotient of n and b. For large b and small n, it is easy to see the advantages this circuit would have because it calculates the total number of elements in a row, and it does all rows in parallel. The more rows there are relative to columns the more advantage provided by this method. There is a problem when n gets too big. When the i-th bit of a given row is sent to the SLFA for counting, the SLFA performs $\max(i) + 1$ many iterations. If n is too large this method will present diminishing returns, as i approaches n. However, for this case there is an alternative. The number of summands can be reduced by half in a fixed number of steps. The method reduces the addition of n summands to the sum of $\max(k) + 1$. Thus, 8 summands can be reduced to $\max(8) + 1 = 4$ summands. If the number of summands is a multiple of 8, then this fact can be used to reduce by half the number of summands. There is another way to reduce summands by half because 4 summands are reduced to $\max(4) + 1 = 3$ summands which in turn are reduced to $\max(2) + 1 = 2$ summands. This means that if the number of summands is a multiple of 4, then the number of summands can be reduced to half in this manner. These alternate methods of reduction into half are achieved by rearranging the vertical and horizontal connections of the grid. Depending on the quotient and size of n and b there will be an optimal size for reduction of summands that minimizes time complexity, and the topology of the nodes is unchanged.

There are several benefits in using this architecture for matrix multiplication. Examples of current solutions to matrix multiplication are proposed and referenced in [Zhang(2013)]. In the case that $n = b$ there are advantages to be exploited for matrix multiplication. Suppose you wish to multiply two $n \times n$ matrices and the entries of the matrices are $n/2$-bit numbers. Storing the two matrices requires $2n^3$ memory units in a rectangular arrangement. A logic grid of $2n^3$ nodes with the same rectangular form can be super-positioned on top of the memory elements. This allows calculation of the dot product of one row and one column in the time it takes to multiply two $n/2$-bit numbers, plus the time it takes to add n-many n-bit numbers. In matrix multiplication it can often be the case that the number b of bits of the entries, is smaller than the numbers of rows and columns. Adaptations can be made for these cases also, based on area-specific use and pipeline needs. This rectangular design of subunits and parallel connections solves some of the basic problems with In-Situ computing [Wang(2023)]. Given the fact that memory is hardwired in rectangular grids, but the logic for computing has many complicated patterns and irregular connections, it is difficult to reconcile both designs in the same space. Von Neumann Arhitecture considers memory and the ALU to be two separate parts of a CPU, at the cost of having to transfer data back and forth between memory and logic units. In this proposal, the memory units can be placed on a bottom layer, and the logic circuitry can be placed on a top layer. This superposition of two rectangular grids of equal size provides a solution to some the problems related with Computing-In-Memory. The delay and energy saved by this architecture merits further investigation and comparison [Hennessy(1990)] to other architectures.

2.4 Power as a Generalization of Product

It is easy to see how the relations between the operations of addition and multiplication can be generalized. The composition powers of $\oplus 1$ are the functions $\oplus n$ given by $\oplus n(x) = \oplus 1^n(x)$. The composition powers of $\oplus x$ are the functions $\odot n$ defined by $\odot n(x) = (\oplus x)^n(0)$. The power function will be defined similarly in terms of operation functions $*n$ such that $x^n = *n(x)$. The function $*n$ is defined by $*n(x) = (\odot x)^n(1)$. In [Ramirez(2019)], there is a description of subtraction, division and powers of set numbers. Here, an alternative definition is given for multiplication and powers. Recall that the multiplication of two sets is given by adding all the sets of the form $\{a \oplus b\}$, where $a \in A$ and $b \in B$. Given any $a \in A$ and $b \in B$, consider the function $f : \{0,1\} \to (A \cup B)$ such that $f(0) = a$ and $f(1) = b$. Then it is true that

$$A \odot B = \bigoplus_{f:\{0,1\}\to(A\cup B)} \{f(0) + f(1)\},$$

where the index f of the sum is taken over every possible function $f : \{0,1\} \to (A \cup B)$ such that $f(0) \in A$ and $f(1) \in B$. Recall that the set of numbers smaller than a fixed number is $2^n - 1 = \{0, 1, 2, \ldots, n-1\}$. The generalized product $(A_1 \odot A_2 \odot \cdots \odot A_n)$ can be written as

$$\bigodot_{i\in 2^n-1} A_i = \bigoplus_{f:2^n-1\to A} \left\{ \bigoplus_{i\in 2^n-1} f(i) \right\},$$

where $A = \bigcup_{i\in 2^n-1} A_i$ and the index f is taken over every function such that $f(i) \in A_i$. It is the addition of singletons, and each singleton is the sum of all the objects in the image of some function f of the index. The commutative and associative properties are trivial to prove from this definition. Changing the order for the multiplication of the sets only changes the order in an addition of sets. This equality gives the expected particular cases. It is easy to see that if $A_k = 0$, for some k, the product is 0. This is true because the sum over the index f is empty; there is no function f such that $f(k) \in A_k$. Furthermore, if all the $A_i = X$ are equal to the same number, the result is X^n.

$$X^n = \bigoplus_{f:2^n-1\to X} \left\{ \bigoplus_{i\in 2^n-1} f(i) \right\}.$$

This expression can easily be verified to satisfy the particular cases. For example, if $n = 1$, then $2^n - 1 = 1 = \{0\}$. What are all the functions of the form $f : \{0\} \to X$? Obviously they are the functions of one component, that select the objects of X. Listing them is easy. For every object $x \in X$, the function f_x defined by $f_x(0) = x$ is considered. The sum of the objects in the image is x, for every function f_x. Adding all the sets corresponding to the addition of the image, taken over every function, gives $X = \bigoplus_{x\in X}\{x\}$. It is easy to see that if $X = 1 = \{0\}$ then there is exactly one function $f : 2^n - 1 \to X$, and it is trivially defined by $f(i) = 0$ for every $i \in 2^n - 1$. This means the result is $1^n = \{0\}$. For $X = 2\{1\}$, again there is exactly one function but this time $f(i) = 1$ for every $i \in 2^n - 1$. Therefore, $2^n = \{n\}$. Up until now, 2^n was just a symbol for denoting the set number $\{n\}$, but now it has acquired its traditional meaning. Now, to define X^0 observe two things. The first is that $2^0 = 1$, and the second is that X^0 is undefined with this definition. The number $2^0 - 1 = 0 = \emptyset$ is the empty set so that there are no functions $f : \emptyset \to X$. Therefore it is justified to define $X^0 = 1$.

2.5 Integers

The structure of integers is not necessary to construct the structure of real numbers. However, a construction of $\mathbb{Z}$ is provided because it introduces methods and concepts of previous and later sections. Operation functions and their inverse functions are used to describe integers. A positive integer $\mathbf{n} \in \mathbb{Z}$ is an operation function $\oplus n$, while its negative integer **-n** $\in \mathbb{Z}$ is the inverse function $(\oplus n)^{-1}$. Notice one important fact. Negative integers can easily be distinguished from positive integers. A negative integer is a function of the form **-n** $: \{n, n \oplus 1, n \oplus 2, \ldots\} \to \mathbb{N}$, while a positive integer is a function of the form $\mathbf{n} : \mathbb{N} \to \{n, n \oplus 1, n \oplus 2, \ldots\}$. This will have to be considered when defining addition of integers; it does not represent any difficulty but the reader must be careful. The integer $\mathbf{0}$ is the identity function of $\mathbb{N}$. The set of negative integers will be represented with the symbol $-\mathbb{N}$. It will be said that $X \subset \mathbb{Z}$ is a *non negative subset of* $\mathbb{Z}$ if $-\mathbb{N} \cap X = \emptyset$, and the like.

The sum of integers is defined in the obvious way, using composition. Let $\mathbf{m} = \oplus m$ and $\mathbf{n} = \oplus n$ positive integers. The composition of these is a positive integer. Define the addition of two positive integers by the relation $\mathbf{m+n} = \oplus m \circ \oplus n$. The sum, **-m-n**, of negative integers $\textbf{-m} = (\oplus m)^{-1}$ and $\textbf{-n} = (\oplus n)^{-1}$, is defined as the composition of inverse functions $(\oplus m)^{-1} \circ (\oplus n)^{-1} = (\oplus n \circ \oplus m)^{-1}$. Given commutativity $\oplus n \circ \oplus m = \oplus m \circ \oplus n$, it follows that **-m-n** is equal to the negative integer $(\oplus m \circ \oplus n)^{-1} = \textbf{-(m+n)}$. The sum of one negative integer **-m** and one positive integer **n** is defined as follows. There are two possible cases. If the corresponding natural numbers satisfy $m < n$, there is a natural number x such that $n = m + x$. Define $\textbf{-m+n} = \mathbf{x}$, where $\mathbf{x} = \oplus x : \mathbb{N} \to \{n - m, n - m + 1, n - m + 2, \ldots\}$. In the contrary case that the natural numbers satisfy $n < m$, then $m = n + x$ for some natural number x. Define addition of these integers by $\textbf{-m+n} = \textbf{-x}$, where $\textbf{-x} = (\oplus x)^{-1} : \{m - n, m - n + 1, m - n + 2, \ldots\} \to \mathbb{N}$. The order relation between m, n determines if **-m+n** is a positive integer or a negative integer. In both cases, the relation $\textbf{-m+n} = (\oplus m)^{-1} \circ \oplus n$ holds. But, how is **n-m** defined? Consider the composition $\oplus n \circ (\oplus m)^{-1}$. In both cases, $m < n$ or $n < m$, the composition is $\oplus n \circ (\oplus m)^{-1} : \{m, m + 1, \ldots\} \to \{n, n + 1, \ldots\}$. Although $\oplus n \circ (\oplus m)^{-1}$ is a well defined composition, it is not an integer. The functions $\oplus n \circ (\oplus m)^{-1}$ and $(\oplus m)^{-1} \circ \oplus n$ are not the same function. However, in the intersection of the domains, these compositions are equal functions. Thus, defining the sum of integers as commutative, $\textbf{n-m} = \textbf{-m+n}$, is justified. To prove addition of integers is associative, let $\mathbf{x}, \mathbf{y}, \mathbf{z}$ integers. Eight different cases have to be proven. The different combinations for x, y, z being positive or negative. Suppose first, y is positive. Then, $\mathbf{x+y} = \oplus x \circ \oplus y$. Consider two sub cases. If z is positive, the associative property holds for $\mathbf{(x+y)+z} = \mathbf{x+(y+z)}$ because the associative property holds for composition of functions. Suppose z is negative. Then $\mathbf{(x+y)+z} = \mathbf{z+(x+y)} = \mathbf{z+(y+x)} = \mathbf{(z+y)+x} = \mathbf{x+(z+y)} = \mathbf{x+(y+z)}$. Going back to the assumption of y, now suppose y is negative and x is positive. The equality $\mathbf{(x+y)+z} = \mathbf{(y+x)+z} = \mathbf{y+(x+z)} = \mathbf{y+(z+x)} = \mathbf{(y+z)+x} = \mathbf{x+(y+z)}$ holds. If $\mathbf{x}$ and $\mathbf{y}$ are negative, then $\mathbf{x+y} = \oplus x \circ \oplus y$. This implies $\mathbf{(x+y)+z} = (\oplus x \circ \oplus y) \circ \oplus z = \oplus x \circ (\oplus y \circ \oplus z) = \mathbf{x+(y+z)}$. This proves addition of integers is associative. The addition of integers **5-3** is equal to the function $\oplus 2$, while the result of **3-5** is $(\oplus 2)^{-1}$.

Ordering integers is natural, in this context. Two integers $\mathbf{x}, \mathbf{y}$ satisfy the inequality $\mathbf{x} < \mathbf{y}$ if and only if $\mathbf{x}(n) < \mathbf{y}(n)$, for any $n \in \mathbb{N}$. For example, $\textbf{-5} < \mathbf{2}$ because $\textbf{-5}(5) = 0 < 7 = \mathbf{2}(5)$. Of course, the order is well defined so that there is no natural number n such that $\mathbf{2}(n) < \textbf{-5}(n)$. To prove $\textbf{-6} < \textbf{-3}$ a set number in the domain of **-3** and **-6** is chosen. Say, the number 6. Then, $\textbf{-6}(6) = 0 < 3 = \textbf{-3}(6)$.

3 Full Version of this Article

The complete version, including the appendixes, can be found in the Supplementary Materials section of the present article.

4 Conclusions

The importance of the axiomatic base is usually undermined because it does not bring any new results or methods into most practical areas of mathematics. Instead, the axiomatic base of mathematics is seen as a *stone in the path*; an obstacle to be dealt with and forgotten. The natural number system proposed allows for natural constructions of classic structures of mathematics. Finite groups are described using natural numbers. Finding all finite groups of n objects is still not trivial but a better notion of attacking this problem is acquired. A minimum set of independent equations that defines each group is obtained in the process. Two groups are isomorphic if their canonical block forms are identical. The set of all finite groups is totally and linearly ordered. This linear order on finite groups is well behaved with respect to cardinality and other aspects. In particular, the commutative group $\mathbb{Z}_n$ is the smallest group of n objects; $\mathbb{Z}_n < G$ for every group G such that $|G| = n$. If $n = p_1^{n_1} p_2^{n_2} p_3^{n_3} \cdots p_k^{n_k}$ is the prime factorization of n, then the commutative group $\mathbb{Z}_{p_1}^{n_1} \oplus \mathbb{Z}_{p_2}^{n_2} \oplus \mathbb{Z}_{p_3}^{n_3} \oplus \cdots \oplus \mathbb{Z}_{p_k}^{n_k}$ is the largest commutative group of n objects. This last behavior was not treated with detail, and is left for future work. Finite groups are also ordered internally. The elements of any finite group are ordered through the canonical naming functions. A criteria for defining equivalent objects of a fixed finite group is obtained, that provides the automorphisms of the group. The set theory for natural numbers was extended to describe infinite mathematical objects such as real numbers, real functions, real valued matrices, sets of real numbers, and structures derived from those, etc. Results pursued in future work can include a thorough description of groups, rings, fields and linear spaces, in the finite and infinite cases separately. Another line of work will include a more comprehensive description of the calculus of real numbers. The theory of types and the Continuum Hypothesis can be considered for future work. There are a variety of ways for coding the information of mathematical structures. Natural data types for the basic structures have been provided, although this library of types must be completed. Trees are used to represent any type of mathematical object. The general procedure for expressing mathematical objects using the smallest type possible is described.

The computational aspects can also be treated with detail, focusing on physical models to represent the arithmetic of Energy Levels. In [Magidor], the author mentions the possibility that *"...we will be able to compare between different Set Theories according to what type of mathematical hinterland they provide for theoretical Physics."* Aside from classic computational schemes that can be improved, such as the one proposed for a simple and linear fast adder, modern computational schemes can also be explored. Encoding and storing mathematical objects (structures of information), is an option to be considered for future work. On the other hand, the linear sum of two waves, in phase, with equal wavelength and frequency, is equal a wave with double the amplitude. The linear superposition of constructive interference from two coherent sources satisfies the numeric principle for addition, $2^n + 2^n = 2^{n+1}$. Thus, measurements on the amplitude of waves can be used as a computational arithmetic model. This could provide a valid approach, for a linear optical computing scheme. Most recently, in [Miscuglio(2020)], it has been noted that *"...the wave nature of light and related inherent operations such as interference and diffraction, can play a major role in enhancing computational throughput..."* And that *"In this view, photons are an ideal match for computing node-distributed networks."* An implementation of the finite-state machine of addition can be a system of coherent wave sources.

Funding

This research received no external funding.

Acknowledgments

Special Thanks to my Professors at undergraduate school. I am specifically thankful to Ms. Sofia Ortega Castillo who has helped me to prepare and organize this material, for talks given at the 52nd National Congress of Mathematics (2019, Monterrey, México), and encouraged me to pursue publication of the material. I am indebted to my professor in group theory, Alonso Castillo Ramírez who has assisted me with all kinds of questions that came up during the time I wrote some of the details on groups. And, to my professor in analysis and probability theory, Victor Pérez Abreu Carreón who has always been a great teacher and friend, whose conversations and classes have inspired a great deal of the work I have tried to carry out. Any corrections or changes to be made are sole responsibility of the author.

Conflicts of Interest

The author declares no conflict of interest.

References

[Corry(2010)] Leo Corry. David Hilbert and the Axiomatization of Physics (1898–1918): From Grundlagen der Geometrie to Grundlagen der Physik. *Springer Netherlands, **2010***

[Benacerraf(1965)] Benacerraf, Paul. What Numbers Could Not Be; *Philos. Rev. **1965**, 74.*

[Thiele(2003)] Rüdiger Thiele. Hilbert's Twenty-Fourth Problem. *The American Mathematical Monthly, 110:1, 1-24, **2003**.* DOI: 10.1080/00029890.2003.11919933

[Ramirez(2019)] Ramírez, J.P. A New Set Theory for Analysis; *Axioms **2019**, 8, 31.*

[Uma(2012)] R. Uma, Vidya Vijayan, M. Mohanapriya, Sharon Paul. 2012. Area, Delay and Power Comparison of Adder Topologies. *International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1, **February 2012**.*

[Singh(2009)] R.P.P. Singh, Parveen Kumar, Balwinder Singh. Performance Analysis Of Fast Adders Using VHDL. *2009 International Conference on Advances in Recent Technologies in Communication and Computing. IEEE Computer Society.*

[Lutz(1994)] D. R. Lutz, D. N. Jayasimha. The Power of Carry Save Addition. *Department of Computer and Information Science, The Ohio State University. **1994**.*

[Sun] Yiqiu Sun, Haichao Yang, et. al. ASIC Design for Bitcoin Mining. *University of Michigan.*

[Wang(2023)] Chenyu Wang, Ge Shi, Fei Qiao, Rubin Lin, Shien Wu and Zenan Hu. Research Progress in Architecture and Application of RRAM with Computing-In-Memory. *Nanoscale Adv., **2023**, 5, 1559-1573.*

[Hennessy(1990)] Hennessy, J.L. and Patterson, D.A. Computer Architecture: A Quantitative Approach.*Morgan Kaufmann, Waltham.**1990**.*

[Lovyagin(2021)] Lovyagin, Yuri N., and Lovyagin, Nikita Yu. Finite Arithmetic Axiomatization for the Basis of Hyperrational Non-Standard Analysis; *Axioms 10, no. 4: 263. **2021**.* https://doi.org/10.3390/axioms10040263

[Bernays(1991)] Bernays, Paul. Axiomatic Set Theory; *Dover: New York, NY, USA, **1991**.*

[Ackermann(1937)] Ackermann, W. Die Widerspruchsfreiheit der allgemeinen Mengenlehre. *Math. Ann. 114, 305–315.*

[Ladner and Fischer(1980)] R. E. Ladner and M. J. Fischer. Parallel Prefix Computation; *Journal of the ACM, 27(4), pp. 831-838, October **1980**.*

[Metropolis, Rota and Tanny(1980)] Metropolis, N.; Rota, G.C.; Tanny, S. Significance Arithmetic: The Carrying Algorithm; *Journal of Combinatorial Theory, Series A, **1973**, 14, 386–421.*

[Abrar(2019)] Abrar, M., Elahi, H., Ahmad, B.A. et al. An area-optimized N-bit multiplication technique using N/2-bit multiplication algorithm. *SN Appl. Sci. 1, 1348 **(2019)**.*

https://doi.org/10.1007/s42452-019-1367-6

[Emmart(2011)] Niall Emmart and Charles C. Weems. High Precision Integer Multiplication with a GPU Using Strassen's Algorithm with Multiple FFT Sizes. *Parallel Processing Letters, Vol.21, No. 03, pp. 359-375 **(2011)**.* https://doi.org/10.1142/S0129626411000266

[Taib(2020)] Muhammad Ikmal Mohd Taib, Muhammad Najmi Zikry Nazri, et. al (2020). Design of Multiplication and Division Operation for 16 Bit Arithmetic Logic Unit (ALU). *JOURNAL OF ELECTRONIC VOLTAGE AND APPLICATION VOL. 1 NO. 2 **(2020)**, 46-54.* DOI: https://doi.org/10.30880/jeva.2020.01.02.006

[Rahman(2013)] Mohammed Ziaur Rahman. Parallel Self-Timer Adder (PASTA). *United States Patent Application, **May 9, 2013**.*

[Franklin(1994)] Franklin, M.A. and Pan, T. (1994) Performance Comparison of Asynchronous Adders. *Proceedings of IEEE Symposium on Advanced Research in Asynchronous Circuits and Systems, Salt Lake City, 3-5 November* ***1994****, 117-125.* https://doi.org/10.1109/ASYNC.1994.656299

[Zhang(2013)] Ting Zhang, Cheng Xu, Tao Li, Yunchuan Qin and Min Nie. An Optimized Floating-Point Matrix Multiplication on FPGA. *Information Technology Journal, 12:* ***2013*** *1832-1838.* DOI: 10.3923/itj.2013.1832.1838

[A'Campo(2003)] A'Campo, N. A Natural Construction for the Real Numbers. *arXiv,* ***2003****;* arXiv:math.GN/0301015 v1.

[Arthan(2004)] Arthan, R.D. The Eudoxus Real Numbers. *arXiv,* ***2004****;* arXiv:math/0405454.

[De Bruijn(1976)] De Bruijn, N.G. Definig Reals Without the Use of Rationals; *Koninkl. Nederl. Akademie Van Wetenschappen: Amsterdam, The Netherlands,* ***1976****.*

[Knopfmacher and Knopfmacher(1988)] Knopfmacher, A.; Knopfmacher, J. Two Concrete New Constructions of the Real Numbers. *Rocky Mt. J. Math.* ***1988****, 18, 813–824.*

[Magidor] Magidor, Menachem. Some Set Theories are More Equal. Preliminary Draft.

[Miscuglio(2020)] Miscuglio, Mario. Photonic Tensor Cores for Machine Learning. *Appl. Phys. Rev.* 7, 031404 (2020); https://doi.org/10.1063/5.0001942