# A Calculus of Number Based on Spatial Forms

by

Jeffrey M. James

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science in Engineering

University of Washington

1993

Approved by____________________________________
(Chairperson of Supervisory Committee)

Program Authorized
to Offer Degree____________________________________

Date____________________________________

University of Washington

Abstract

# A Calculus of Number Based on Spatial Forms

by Jeffrey M. James

Chairperson of Supervisory Committee: *Professor Judith Ramey*
*Technical Communication*

A calculus for writing and transforming numbers is defined. The calculus is based on a representational and computational paradigm, called *boundary mathematics*, in which representation consists of making distinctions out of the void. The calculus uses three boundary objects to create numbers and covers complex numbers and basic transcendentals. These same objects compose into operations on these numbers. Expressions transform using three spatial match and substitute rules that work in parallel across expressions. From the calculus emerge generalized forms of cardinality and inverse that apply identically to addition and multiplication. An imaginary form in the calculus expresses numbers in phase space, creating complex numbers. The calculus attempts to represent computational constraints explictly, thereby improving our ability to design computational machinery and mathematical interfaces. Applications of the calculus to computational and educational domains are discussed.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# PREFACE

This work attempts to clarify mathematics, to make complex ideas accessible to those with less mathematical training. In this thesis, I have reduced number and operator representation down to three forms, revealing a simplicity to numbers. I hope this work may aid those who have struggled to understand numbers and computation, as well as elementary algebra, for I surely consider these topics in new light.

I am not a mathematician, so writing a thesis about mathematics has been a challenge for me. For years, I have struggled to understand what research in mathematics is all about that I might contribute to its continuing progress. Alas, essential concepts continue to elude me and I remain uncertain about what mathematicians are actually doing. Nonetheless, I believe my work to have some relevance to them and have tried to describe it in a manner that would avail mathematical researchers to make use of it, though my audience also includes the less mathematically inclined.

I have benefitted from many discussions with friends and colleagues during the development of this material. Most recognize some insight here, though I am not certain whether they liked the material itself or just the idea of trying to make mathematics easier to understand.

I thank William Bricken with my heart and soul. William introduced me to boundary mathematics and has served as my mentor throughout this research—his guidance and direction has made it all happen.

I thank the rest of my thesis committee: Judith Ramey, William Winn, and Steven Tanimoto. Each of them played unique roles in the development of this material. I was privileged to have had such excellent support for my research.

I credit my zest for research to Penelope Sanderson, with whom I studied as an undergraduate at the University of Illinois. Penny challenged me with interesting research problems and gave me opportunities to present my findings to receptive audiences. I am forever grateful to her.

I thank the students and staff of the Human Interface Technology Laboratory and its director, Dr. Thomas A. Furness, for providing a stimulating environment for developing these ideas. I thank Kimberley Osberg, Dav Lion, and Greg Woodward in particular for their personal support. I thank US West for funding part of this research and I thank all members of the consortium for supporting the lab.

Though I wrote this document at the HIT Lab in Seattle, I "polished the turd" at Interval Research in Palo Alto. I thank Dick Shoup, the Natural Computing group, and everyone at Interval for their support of my research.

I extend my warmest appreciation to Ann Miller, with whom I shared the joy of discovering many of the ideas herein. Ann taught me humility and compassion and will always be in my heart.

Finally, I thank Laura Grant, the love of my life. I could never have finished this thesis without her intellectual and emotional support. Though it took a while, I finally managed to "swallow the toad!"

Jeffrey Mark James
Human Interface Technology Laboratory
October 14, 1993

*To my very first mathematics teacher, my father.*

Chapter 1

# INTRODUCTION

Numbers are simpler than they appear. This thesis presents a calculus of number that demonstrates this simplicity.

We commonly represent and manipulate numbers with a centuries-old notation comprised of Arabic digits, decimal point, plus sign, minus sign, times sign, fraction bar, a few other operators, and some constants. Typical numbers are 0, 1, 2, 10, $-1$, $3/7$, 2.9, $\sqrt{2}$, $e^2$, $i\pi$. The standard Arabic notation represents ten numbers directly as digits and a few more as symbolic constants. All others are constructed out of these using arithmetic operations. Many of the operations used to build numbers are implicit in the representation, such as the magnitude gain in place value. Hiding information in this manner achieves conciseness but leaves the forms abstract and dissociated from their own behavior.

These traditional representations are but one embodiment of numbers. Formal alternatives exist but generally lack the scope and conciseness of the standard form. Other representations are useful because they display interesting properties of numbers. Conway Numbers, for instance, build real numbers out of a single object, the bi-set [12]. This thesis presents a representation of numbers, called *boundary numbers*, in which numbers and arithmetic operations are built out of three objects.

## 1.1 Minimalism

The calculus in this thesis progresses representation of number towards a minimal basis. It begins with the most fundamental acts of representation. By starting from scratch, the calculus avoids the conceptual baggage that has been introduced into standard notation throughout its development.

The calculus is an attempt to minimize the *implicit* constraints on form manipulation to just those necessary to define the system of mathematics. Other constraints should be *explicit* in the forms. Commutativity, for example, would be explicit if commutative functions displayed symmetry between their arguments and

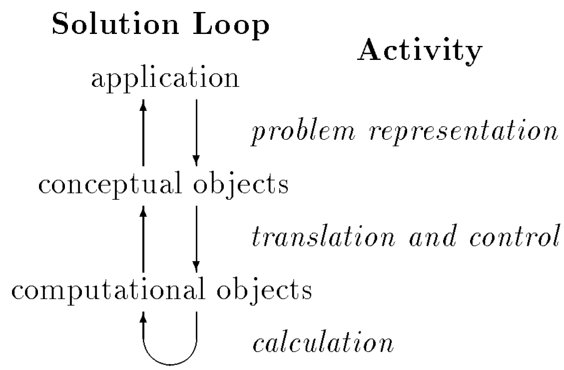



Figure 1.1: Stages of Mathematical Representation.

non-commutative functions did not. By making constraints explicit, much "knowledge" of mathematics can be embodied in the representation.

Standard notation has abstracted away the behavior of its forms so that knowledge of their behavior has become implicit. For example, the notation abstracts away magnitudes of the digits, making $1 + 1 = 2$ not altogether obvious but requiring memorization. A model of this disembodiment is shown in Figure 1.1. The standard notation represents the *conceptual* objects of numbers and functions but not the *computational* objects upon which constraints are imposed.

When using math, we work through the solution loop shown in Figure 1.1. We represent elements of a problem as conceptual objects in a mathematical notation. We manipulate these objects according to calculation rules, usually through separate computational objects. Guided by solution techniques, we manipulate the conceptual objects into forms that lend insights into the problem.

Layers of mathematical objects are necessary because most mathematical forms cannot be "directly" manipulated with a concise set of rules. Comprehensively defined manipulation often requires intermediate forms. Moreover, these intermediate forms can be completely disparate from the original form, making the translation itself complicated.

For example, with digit magnitude implicit, numbers are often added by use of separate representations. Many computers use bits as computational objects because bits can be physically added by digital circuitry. Many people learn to add by using

Table 1.1: Simple Calculations in Standard and Boundary Forms.

| **Standard** | **Boundary** |
|---:|---|
| $1+1=2$ | ○○=○○ |
| $21+11=32$ | ([$b$][○○])○([$b$][○])○=([$b$][○○○])○ |
| $2\times 3=6$ | ([○○][○○○])=○○○○○○ |
| $3^2=3\times 3$ | (([[○○○]][○○]))=([○○○][○○○]) |
| $x+0=x$ | $x$=$x$ |
| $x\times 1=x$ | ([$x$][○])=$x$ |
| $x\times 0=0$ | ([$x$][])= |
| $x^1=x$ | (([[$x$]][○]))=$x$ |
| $x^0=1$ | (([[$x$]][]))=○ |
| $1/x=x^{-1}$ | (<[$x$]>)=(([[$x$]][<○>])) |
| $x^y x=x^{y+1}$ | ([(([[$x$]][$y$]))][$x$])=(([[$x$]][$y$○])) |

objects which carry the mathematical constraints explicitly, such as counting blocks. The counting blocks serve as computational objects upon which calculation can be directly performed. Because standard notation does not embody the transformation constraints, surrogate representations are necessary for machines and children, both of which require explicit declarations.

Standard notation conceals mathematical behavior in ways besides hiding digit magnitudes. The equations on the left side of Table 1.1 collectively reflect mathematical knowledge which, when known, make the equalities obvious. Visually, they are not obvious: the features of the forms provide no clues as to their manipulative constraints. The forms carry none of this knowledge, a failure which places unnecessary demands on users of mathematics.

The boundary calculus shifts the focus of representation from the conceptual objects of numbers and functions to the computational objects where the dynamics of the forms can be directly captured. It makes a clear separation between the fundamental constraints on number manipulation and the concepts that direct this manipulation. The equations on the right side of Table 1.1 are the same equalities translated to the boundary calculus. Here, $b$ represents the base of the place-value system, i.e. $b \doteq$ ○○○○○ ○○○○○. The expressions are longer because they are constructed out of very basic objects, built into structures that reveal information about their own transformational constraints. The three axiomatic transformations given in the next section are all that is required to demonstrate the boundary equalities.

The mechanics of number manipulation are simple in the boundary calculus. More complicated concepts are not "built-in" but instead appear as directed use of it. Techniques for managing numbers, such as separating magnitude or naming small integers, exist independent of the axioms of this system. The calculus provides a crisp mathematical substrate within which to construct and illustrate mathematical concepts that were previously interwoven with the symbolic rules themselves.

## *1.2 The Calculus*

This thesis defines a few mathematical forms and how they transform, i.e. it defines a *calculus*. These forms represent *numbers*, as opposed to logic or sets. And these forms are based on a representational paradigm that is *spatial*. In short, this thesis defines a calculus of number based on spatial forms.

This material assumes a minimalist approach to representation called *boundary mathematics* [3, 17]. The ideas were initiated by G. Spencer-Brown [11] and have been extended in the domain of numbers by Spencer-Brown himself, as well as Louis Kauffman [22, 24] and William Bricken [5, 9]. This work extends theirs by introducing general forms for cardinality and inverse, and by defining phase. This work also gives a functional interpretation of boundaries not found elsewhere, treating them as logarithmic functions.

In boundary mathematics, representation begins with empty space, a complete lack of structure, *void*. Upon this void, *distinctions* are made that impose structure and intention. Distinction may be thought of as a *boundary* that delineates space. Boundaries may be nested and collected to form different configurations. A configuration of boundaries represents a mathematical expression that can be transformed according to *match and substitute* rules, as defined for the mathematical system [11, 3].

The boundary numbers described here use more than one distinction, a practical move which as yet has not been philosophically justified in the paradigm. Multiple boundaries force a distinction that is not a cleaving of the space; instead, it is a differentiation of the boundaries. Here, this distinction is accomplished with geometric characteristics: the boundaries are drawn round $\circ$, square $\square$, or pointy $\triangle$. To hold contents, they are also drawn as delimiters: ($A$), [$A$], <$A$>.

These three boundaries are the fundamental forms of the calculus of number. Alphabetic characters are also used to denote unknown values and constants. Equality is expressed as usual, denoting configurations of equal numerical value.

Natural numbers can be represented by collecting a single boundary, as in a tally system. Call this boundary *instance* and draw it round. Counting proceeds by accumulating empty instance boundaries, as o, oo, ooo,... and so on. These numbers add by collecting them in space, no computation is necessary. A second boundary—call it *abstract* and draw it square—enables formation of multiplication and power functions. For example, the multiplication $2 \times 3$ appears in boundary notation as ([oo][ooo]) and the cube of two, $2^3$, appears as (([[oo]][ooo])).

A third boundary—call it *inverse*—can be used to construct additive and multiplicative inverses, and other inverse structures such as roots. Inverse extends the calculus to the integers by serving as the additive inverse, to the rationals by serving as the multiplicative inverse, and to algebraic irrationals in other combinations. A full sampling of boundary numbers is shown in Table 1.2 and the boundary operations are shown in Table 1.3.

Calculation on these forms is completely defined by three axioms: *involution*, *distribution*, and *inversion*, shown in Table 1.4. Involution defines instance and abstract to be functionally opposite. Distribution defines them to additionally have a distributive relationship. Inversion defines the inverse boundary to behave as an inverse operator. The three axioms govern all transformations of boundary expressions.

The calculus constructs general forms that are not available in the standard notation. Using involution and distribution, a repeated form such as $AAA$ can be transformed into an equivalent expression that requires only one reference to that form, as ([$A$][ooo]). This transformation is called *cardinality* and is shown in Table 1.5. This means of counting applies to repeated addition and to repeated multiplication; it is therefore considered to be a generalized form of cardinality.

The inverse boundary is also unparalleled in standard notation. It is used to form the additive inverse as well as the multiplicative inverse (see Table 1.3). All of the properties of this generalized inverse apply to both of these contexts. These properties include collection of inverses, cancellation with itself, and relocation, all shown in Table 1.5.

Table 1.2: Mapping of Numbers

| **Standard** | **Boundary** |
|---:|---|
| $0$ | |
| $1$ | ○ |
| $2$ | ○○ |
| $-1$ | <○> |
| $-2$ | <○○> |
| $1/2$ | (<[○○]>) |
| $2 \times 3$ | ([○○][○○○]) |
| $2/3$ | ([○○]<[○○○]>) |
| $4\frac{1}{2}$ | ○○○○(<[○○]>) |
| $3^2$ | (([[○○○]][○○])) |
| $\sqrt{3}$ | (([[○○○]]<[○○]>)) |
| $243$ | ([$b$][([$b$][○○])○○○○])○○○ |
| $10^6$ | (([[$b$]][○○○○○○])) |

Table 1.3: Mapping of Functions

| **Standard** | **Boundary** |
|---:|---|
| $a$ | $a$ |
| $-a$ | <$a$> |
| $1/a$ | (<[$a$]>) |
| $a+b$ | $ab$ |
| $a-b$ | $a$<$b$> |
| $a \times b$ | ([$a$][$b$]) |
| $a/b$ | ([$a$]<[$b$]>) |
| $a^b$ | (([[$a$]][$b$])) |
| $a^{-b}$ | (([[$a$]][<$b$>])) |
| $\sqrt[b]{a}$ | (([[$a$]]<[$b$]>)) |

Table 1.4: Axioms of the Calculus.

| | |
|---|---|
| *Involution* | $([A]) \doteq A \doteq [(A)]$ |
| *Distribution* | $(A[BC]) \doteq (A[B])(A[C])$ |
| *Inversion* | $A<A> \doteq$ |

Table 1.5: Theorems of the Calculus.

| | |
|---|---|
| *Cardinality* | $A...A = ([A][\circ...\circ])$ |
| *Dominion* | $\Box A = \Box$ |
| *Inverse Collection* | $<A><B> = <AB>$ |
| *Inverse Cancellation* | $<<A>> = A$ |
| *Inverse Promotion* | $<(A[B])> = (A[<B>])$ |
| *Phase Independence* | $[<(A)>] = A[<()>]$ |
| *J Cancellation* | $[<\circ>][<\circ>] =$ |

Its forms can be interpreted so as to extend the calculus to complex and transcendental numbers. Taking cardinalities of the inverse produces radian values that can act as complex numbers when their manipulation is further restricted. The instance and abstract boundaries act as exponential and logarithmic functions, respectively. By interpreting these to be of base $e$, basic transcendental values and trigonometric functions can be formed. These interpretations are included in Tables 1.2 and 1.3.

## *1.3 Conclusions*

The calculus represents many types of numbers but it covers only part of number mathematics. It lacks larger structures for doing mathematics and requires much practical support to make it useful. There are many additions that can be made to the calculus and areas to which the content can extend. Future work with the calculus will expand and enhance the ideas presented here.

The calculus presents a new paradigm of number representation that challenges how mathematics is currently done. This material may impact the way mathematics is done physically in hardware, logically in software, and conceptually in an interface with mathematics. It may also impact how mathematics is taught, principally because it serves as an alternative to the standard notation.

Though the boundary forms are not immediately comprehensible, familiarity makes them legible. And since many of the transformations are easily visualized, the boundary forms eventually appear quite natural. Recall Table 1.1 in which calculations in standard notation are paralleled in boundary notation. Many of the boundary calculations are immediate, while others require only a few applications of the axioms. Appendix A contains more comparisons and Appendix B contains examples, including a derivation of the quadratic formula. Overall the boundary calculations are much simpler, indicating that standard form makes computation—and therefore understanding—unnecessarily complex.

# Chapter 2

# PRIOR WORK

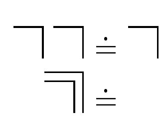
Boundary mathematics was first explored as a paradigm of mathematical representation in the context of propositional logic. It has been applied extensively to logic [11, 3, 31, 1, 19, 32] as well as to other systems of mathematics, including imaginary logic [25, 34, 35, 33], algebra [13, 27], self-reference and recursion [36, 20, 16, 25, 21], control and deduction in computer science [4, 7, 8, 6, 14], and numbers [5, 10, 24].

The applications of boundary mathematics to numbers are often called *boundary numbers*. G. Spencer-Brown, Louis Kauffman, and William Bricken have each created systems of boundary numbers [5, 10, 24, 23, 22]. Their systems share the basic characteristics associated with boundary mathematics but differ in detail. In all of them, boundaries serve as the objects of the system and as operators on those objects. Numbers are created out of boundaries and boundaries build into arithmetic operations. This common basis of form allows calculation directly on the forms, erasing and rearranging to transform an expression. Each of these systems are briefly described using their creator's original notation.

## *2.1 Spencer-Brown Numbers*

The boundary paradigm began with Spencer-Brown's *Laws of Form* in which he reduced logic to a single distinction [11]. He shows how two fundamental choices of reduction, *calling* and *crossing*, produce boolean arithmetic. In his work, Spencer-Brown draws a distinction as a *mark*, ⌉, shown below in these rules.

*Calling*
*Crossing*



His number system attempts to adapt calling and crossing to the number domain. He variously treats space as addition and as multiplication but, for simplicity, only the former system will be considered [10, 5].

Table 2.1: Definition of Spencer-Brown Numbers.

| *Numbers* | *Operators* | *Rules* |
|---|---|---|
| $0 \rightarrow$ | $a + b \rightarrow a\ b$ | $\overline{\overline{\ }\vert}\vert \ \doteq$ |
| $1 \rightarrow \overline{\ }\vert$ | $a \times b \rightarrow \overline{\overline{a}\vert\ \overline{b}\vert}\vert$ | $\overline{\overline{at}\vert\,\overline{bt}\vert \ldots}\vert \ \doteq\ \overline{\overline{a}\vert\ \overline{b}\vert \ldots}\vert\, t$ |
| $2 \rightarrow \overline{\ }\vert\,\overline{\ }\vert$ | $a^b \rightarrow \overline{b}\vert :a$ | $\overline{\overline{a}\vert}\vert \ = \ a$ |
| $3 \rightarrow \overline{\ }\vert\,\overline{\ }\vert\,\overline{\ }\vert$ | | $\overline{\ }\vert :a \ = \ \overline{\ }\vert$ |

With space as addition, the natural numbers are easily formed. The void acts as zero and counting proceeds by accumulating marks: $\overline{\ }|$, $\overline{\ }|\,\overline{\ }|$, $\overline{\ }|\,\overline{\ }|\,\overline{\ }|$, … and so on. Numbers are added by spatial collection whereas multiplication and power operations are composed using marks. Spencer-Brown's numbers are summarized in Table 2.1.

He gives two axioms for calculating on these forms, *universe* and *transfer*. With these axioms he derives many theorems, including *reflexion* and *null power*. Universe and null power are his numerical versions of calling and crossing.

| | | | |
|---|---|---|---|
| *Universe* | $\overline{\overline{\ }\vert}\vert$ | $\doteq$ | |
| *Transfer* | $\overline{\overline{at}\vert\,\overline{bt}\vert \ldots}\vert$ | $\doteq$ | $\overline{\overline{a}\vert\ \overline{b}\vert \ldots}\vert\, t$ |
| *Reflexion* | $\overline{\overline{a}\vert}\vert$ | $=$ | $a$ |
| *Null Power* | $\overline{\ }\vert :a$ | $=$ | $\overline{\ }\vert$ |

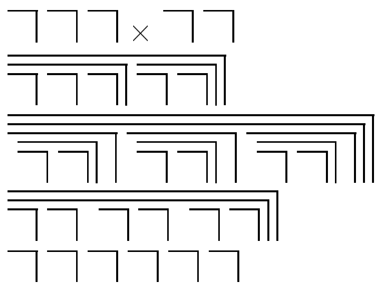
In addition to the mark, he uses a vaguely defined colon to disambiguate conflicting results which would equate $0^1 \rightarrow \overline{\ }|:\overline{\ }|$ and $2 \rightarrow \overline{\ }|\,\overline{\ }|$. This colon does not completely resolve the problems it was introduced for.

These numbers add by spatial collection. For example, $3 + 2 = 5$.

| | |
|---|---|
| $3 + 2$ | Given |
| $\overline{\ }\vert\,\overline{\ }\vert\,\overline{\ }\vert + \overline{\ }\vert\,\overline{\ }\vert$ | Number Rewrite |
| $\overline{\ }\vert\,\overline{\ }\vert\,\overline{\ }\vert\,\overline{\ }\vert\,\overline{\ }\vert$ | Addition Rewrite |
| $5$ | Number Rewrite |

Numbers multiply by the form $a \times b \rightarrow \overline{\overline{a}|\,\overline{b}|}|$. The form reduces by transfer, which distributes the whole of one number throughout the units of the other. For example, $3 \times 2 = 6$.

3 × 2 — Given



Number Rewrite

Multiplication Rewrite

Transfer, $t = \overline{\;\rceil\;\rceil}|$

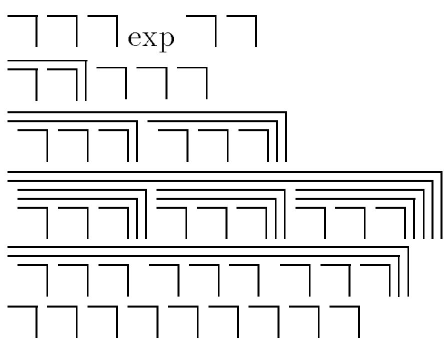
Reflexion (3x)

Reflexion

6 — Number Rewrite

Powers are expressed as $a^b \rightarrow \overline{\text{b}}\rceil\, a$. This form creates a repeated multiplication where the arity is determined by the exponent and the base is introduced as the arguments using transfer. For example, $3^2 = 9$.

$3^2$ — Given



Number Rewrite

Power Rewrite

Transfer, $t = \rceil\rceil\rceil$

Transfer, $t = \overline{\rceil\rceil\rceil}|$

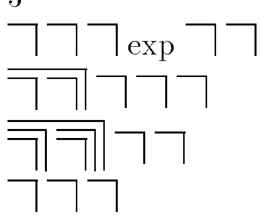
Reflexion (3x)

Reflexion

9 — Number Rewrite

Spencer-Brown's system has two major problems. First, transfer is not fully compatible with spatial match and substitute: it requires that the $t$ match the entire context. For example, the previous problem produces a conflicting result if transfer matches only part of the context.

$3^2$ — Given



Number Rewrite

Power Rewrite

Transfer, $t = \rceil$

Reflexion (2x)

3 — Number Rewrite

The other problem with this system is its treatment of zero. A multiplication by zero should reduce to zero, as $\overline{\rceil\;\text{x}}| = \overline{\rceil}| = \quad$. Similarly, a zero exponent should

Table 2.2: Definition of Kauffman Numbers.

| *Numbers* | | *Rules* | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $0 \rightarrow$ | *Operators* | $<*>$ | $\doteq$ | $**$ | | $>*$ | $=$ | $*>\overline{*}$ |
| $1 \rightarrow *$ | $a + b \rightarrow a \;\; b$ | $< \; >$ | $=$ | | | $>\overline{*}$ | $=$ | $\overline{*}>*$ |
| $2 \rightarrow <*>$ | $a * b \rightarrow a \;\texttt{i}\; b$ | $<\overline{*}>$ | $=$ | $\overline{**}$ | | $><$ | $\doteq$ | |
| $3 \rightarrow <*>*$ | $-a \rightarrow a \;\texttt{i}\; \overline{*}$ | $*\overline{*}$ | $\doteq$ | $\overline{*}*$ | $=$ | $w*$ | $\doteq$ | $*w$ |
| $4 \rightarrow <<*>>$ | | $\overline{\overline{*}}$ | $=$ | $*$ | | | | |

reduce to one, as $\urcorner y = \urcorner$. Spencer-Brown performs both of these reductions using transfer into a mark with nothing inside, as $\urcorner t = \urcorner$. However, this rule undermines addition space because adding one provides an empty $\urcorner$ for obliterating the rest of an expression. The colon was his attempt to disambiguate this paradox but its use was not thoroughly defined.

Spencer-Brown's numbers represent the natural numbers and the system computes basic arithmetic operations on them except for those requiring an inverse. The system is simple and concise, using only a single distinction, but it ultimately breaks down.

## *2.2 Kauffman Numbers*

Kauffman's number system also represents the natural numbers using boundaries. It contrasts from other boundary systems in that it transforms by string matching rules rather than by spatial matching rules.

In his system, a boundary represents magnitude, a doubling of contents. The object $*$ represents a unit and the boundary $<*>$ denotes a magnitude of its contents. Here, it is assumed to be a doubling, $<*> \doteq **$. Counting proceeds in Kauffman numbers as $*$, $<*>$, $<*>*$,... and so on, as a base-2 system. These numbers add by concatenation. They multiply by replacing the units of one number with the entirety of the other, denoted by the insertion $A\texttt{i}B$. Kauffman numbers are summarized in Table 2.2.

Kauffman uses an overbar to represent the additive inverse, making $\overline{*}$ the negative unit. A positive unit and a negative unit cancel out, as $*\overline{*} \doteq \overline{*}* \doteq$ . Many transformation rules involve both the inverse and the doubling boundary. Because this system uses string rewrite rules, ordering *is* significant. The rules on the left of Table 2.2 are for doubling and inverse, whereas the rules on the right handle sequencing.

In this system, numbers add by concatenation, i.e. spatial collection in one dimension. For example, $3+2=5$. The result is transformed to a canonical form using the string rewrite rules.

| | |
|---|---|
| $3+2$ | Given |
| <∗>∗ + <∗> | Number Rewrite |
| <∗>∗<∗> | Addition Rewrite |
| <∗><∗>∗ | $w* = *w$ |
| <∗∗>∗ | >< = |
| <<∗>>∗ | ∗∗ = <∗> |
| 5 | Number Rewrite |

Multiplication is performed by replacing one expression for the units in the other. For example, $3 \times 2 = 6$.

| | |
|---|---|
| $3 \times 2$ | Given |
| <∗>∗ × <∗> | Number Rewrite |
| <∗>∗ i <∗> | Multiplication Rewrite |
| <<∗>><∗> | Multiplication Substitution |
| <<∗>∗> | >< = |
| 6 | Number Rewrite |

Subtraction is performed by multiplying by the negative unit and then adding. The result is reduced using the rewrite rules with the negative unit. For example, $8-1=7$.

| | |
|---|---|
| $8-1$ | Given |
| $<<<*>>> \; - \; *$ | Number Rewrite |
| $<<<*>>> \; + \; \overline{*}$ | Subtraction Replacement |
| $<<<*>>>\overline{*}$ | Addition Rewrite |
| $<<<*>>\overline{*}>*$ | $>\overline{*} \; = \; \overline{*}>*$ |
| $<<<*>\overline{*}>*>*$ | $>\overline{*} \; = \; \overline{*}>*$ |
| $<<<*\overline{*}>*>*>*$ | $>\overline{*} \; = \; \overline{*}>*$ |
| $<<<>*>*>*$ | $*\overline{*} \; =$ |
| $<<*>*>*$ | $<> \; =$ |
| $7$ | Number Rewrite |

Though this system is limited to the integers, Kauffman has defined another boundary as the opposite of the magnitude boundary, such that $(<A>) \doteq <(A)> \doteq A$. The string rewrite rules for this system are quite extensive due to the many permutations of the characters and are not included here.

Alternatively, the extended Kauffman numbers can be defined spatially with three rules. The definition below uses his delimiter form of the inverse rather than the overbar form shown above [24].

| | |
|---|---|
| *Double* | $AA \doteq <A>$ |
| *Inverse* | $A[A] \doteq$ |
| *Half* | $(<A>) \doteq <(A)> \doteq A$ |

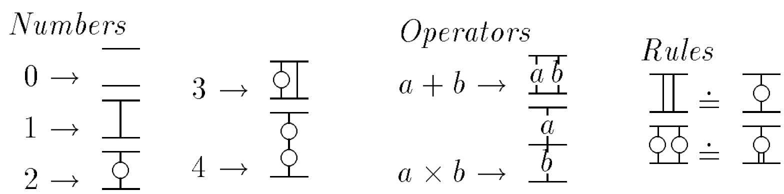
Kauffman numbers represent the integers and the system computes basic arithmetic operations except division and powers. Addition by collection is immediate, as is multiplication by substitution. Reduction is done locally by string rewrite rules.

## 2.3 *Bricken Numbers*

Kauffman's work was given a different twist by William Bricken, who interpreted his basic forms as networks [5]. The network form provides singular references and a clear representation of the structures being manipulated. Of particular benefit, singular reference allows multiplication by stacking, providing an algebraic form. Bricken numbers are summarized in Table 2.3.

Bricken numbers are networks whose value is determined by their connectivity. The network includes a context and ground, shown as lines above and below the

Table 2.3: Definition of Bricken Numbers.




network. Calculation changes this connectivity to a canonical form.

Bricken deviates from Kauffman's forms by implementing inverses as gradients in the connections. He extends the system with algebraic variables and canonical equations and uses these extensions to derive algebraic solutions.

As with Kauffman numbers, Bricken's do not support any exponential form. The great advantage to Bricken numbers is that they clearly define calculation as changes in connectivity.

## *2.4 Conclusions*

Each of the systems described uses distinction to build numbers. The representations compute in-place by the transformation rules of each system. Each operates in parallel, allowing computation to occur locally within a vast configuration.

Each of these three systems cover addition and multiplication of natural numbers. Spencer-Brown's system also has exponentiation, Kauffman's system also has the additive inverse, and Bricken's system has additive and multiplicative inverses. However, none of these are clearly extensible beyond their current scope of coverage. The limited scope of these systems restricts their usefulness as computational foundations. Collectively they suggest that it is possible to construct a boundary system for numbers with broader scope.

# Chapter 3

# BOUNDARY MATHEMATICS

## *3.1 Introduction*

This chapter describes the principles of *boundary mathematics*, upon which this thesis builds the calculus of number.

Boundary mathematics is a representational and computational paradigm for mathematics, based on the concept of distinction. The paradigm was initially described by Spencer-Brown [11] but the term comes from Bricken's interpretation of his work in which a distinction is drawn as a topologically closed boundary[1] [3]. The boundary mathematics paradigm prescribes *forms* of representation as well as their mechanism of *transformation*.

## *3.2 Forms*

Boundary mathematics is based on spatial forms. Representation begins with complete lack of form and structure: empty space or *void*. Structure is added to this void by cleaving the space, by drawing a *distinction*. Each distinction adds structure, cleaving new spaces where further distinctions can be made.

Drawing a distinction is independent of the dimension of the space. For the purposes of representation, one dimensional distinctions are typographical delimiters, two dimensional distinctions are closed loops on a page, and three dimensional distinctions are solid objects.

### *3.2.1 Void*

All forms stand in contrast to lack of form, void. See Figure 3.1. To begin with anything but the void would assume too much about representation. The void imposes

[1] While Spencer-Brown equates the concepts of distinction and boundary, his notation fails to exploit this. Drawn as a boundary, distinction appears more complete than Spencer-Brown's *mark*, ⏋, and can be interpreted in linear form as paired delimiters.

Figure 3.1: The Void.

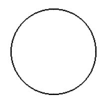


Figure 3.2: A Distinction.

no structure or intention on forms which are made. In contrast, linear, token-based representations impose a great deal of structure.

The void plays a vital role in boundary mathematics. It is anywhere and everywhere; it is where all distinctions are made. The void does not go away: it permeates all boundary representations [3].

Equivalences with the void can be easily mistaken for typographical errors when written linearly. The equivalence form $A = B$ puts no explicit boundary around the expressions being equated so a void equivalence such as *A*<*A*>= may at first appear erroneous. Since void expressions are legitimate in boundary mathematics, these expressions should be recognized as valid.

### *3.2.2 Distinction*

Distinction is the primary form of boundary mathematics. A distinction cleaves space, imposing structure and intention upon it. Distinctions serve both a syntactic and semantic role. The meaning lies in making the separation.

Figure 3.2 shows a distinction in the two-dimensional space of this page. The shape and scale of this distinction are irrelevant. The distinction serves only to separate the *content* of the distinction from the *context* in which it was made. The content and context of a distinction are labeled in Figure 3.3.

Additional distinctions are made with respect to the first distinction. Distinction is an *action*; the result of distinction is an *object* for further action. There are two ways to make a second distinction: in the content or context of the first distinction.

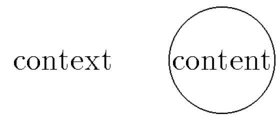



Figure 3.3: Content and Context of a Distinction.

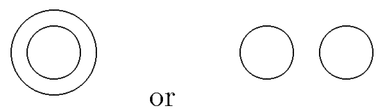



Figure 3.4: Two Choices for Further Distinction.

These choices are shown in Figure 3.4.

Distinctions are drawn in the figures as boundaries. Here, they are written textually as paired delimiters, `()`. Nested distinctions appear as `(())` and collected distinctions appear as `()()`.

### *3.2.3 Multiple Boundaries*

Boundary mathematics uses a single distinction to represent logic [11, 3], but the mathematics in this thesis uses three boundaries. Having different boundaries requires drawing a separate distinction between the boundaries aside from the cleaving of space. The separate distinction between the boundary types is one of convenience, to advance boundary numbers and no philosophical justification will be attempted. In this thesis, a distinction and a boundary are not synonymous since `(...)`, `[...]`, and `<...>` all represent spatial distinctions but a separate orthogonal distinction is made between them that is not a cleaving of space.

## *3.3 Transformation*

The dynamic element of a system of boundary mathematics is how to manipulate forms, how to *transform* boundary expressions. A boundary expression is essentially a network. The connectivity of the network determines its value. Equivalences between connectivities defines the system of mathematics.

### *3.3.1 Match and Substitute*

Equivalences are described by *equivalence rules* which transform by algebraic *match and substitute*. The rules given to define a system are its *equivalence axioms*.

An equivalence rule consists of two or more boundary *templates* which include *template variables*. To apply a rule to a target expression, replace each template variable with an expression so that one of the filled templates *matches* some part of the target expression. Another filled template from the rule can then *substitute* for the matched template in the target expression.

For example, the rule (($A$)) = $A$ includes two templates which use the template variable, $A$. The target expression (((∘∘))∘) can be matched by the left template by replacing the $A$ with ∘∘ to produce ((∘∘))=∘∘. Substituting the right template for the left template in the target expression transforms it into (∘∘∘), essentially erasing two boundaries.

### *3.3.2 Properties*

Transformations in boundary mathematics have some special properties. Match and substitute can operate in parallel and at all levels of granularity in an expression. For example, the rule (($A$)) = $A$ applies to the form (( (()) () )) twice; both applications can occur *simultaneously* without conflict to reduce it to ().

In these boundary numbers, the equivalence rules are bi-directional. The rule itself states no direction of preference, though one may be imposed for reduction purposes. Therefore, a reduced expression is mathematically equivalent to many more complicated forms. Because functions are constructed with the same boundaries as are numbers, the specification of a calculation is equivalent to a reduced result. The specification is the result—transformation just changes the form that it is in.

## *3.4 Conclusions*

Boundary mathematics is a representational and computational paradigm based on spatial forms. Boundary notation looks and acts differently than standard notation, using spatial constraints and spatial properties rather than rote rules.

Spatial forms differ fundamentally from standard forms. Space does not impose ordering constraints; associativity and commutativity are not necessary as no order or arity has been imposed in the first place. Non-representation has meaning and contributes to computation in ways unparalleled in standard systems. The void acts as a built-in identity that is always available. Spatial match and substitute operates at all parts of an expression simultaneously, an inherently parallel computation mechanism. These properties make boundary mathematics a unique and worthwhile paradigm of representation and computation.

Many concepts generally attributed to standard mathematics are notably absent from boundary mathematics and vice-versa. Boundary notation makes no distinction between objects and operations upon them, as both are built out of the same forms. Standard mathematics does not allow void substitution but boundary mathematics does. Boundary rules tend to embody symmetry of function whereas standard mathematics addresses identity elements and rearrangement properties. The tradeoffs are many.

Boundary mathematics is exotic enough to seem obscure and of unclear advantage. This thesis will demonstrate the advantage of the boundary mathematics paradigm by defining in it a calculus that simplifies number representation and calculation.

Chapter 4

# INSTANCE AND ABSTRACT

## 4.1 Introduction

This chapter introduces boundary numbers by defining a two-boundary calculus. The two-boundary calculus maps to natural numbers, complete with addition, multiplication, and power functions. The calculus also builds a structure without parallel in standard notation: a generalized cardinality form that applies to both addition and multiplication.

In this calculus, space is treated as addition. A single distinction represents natural numbers by repeating the distinction for the value. This distinction shall be called *instance* and drawn as a circular boundary when empty "○" or as parentheses otherwise "($A$)." An empty instance forms the natural number one; it is the *unit* for counting. Nestings of instance are not natural numbers; these forms will be characterized in Chapter 6.

A second distinction builds with the first to make multiplication and power functions. This distinction shall be called the *abstract* and drawn as a square boundary when empty "□" or as square brackets otherwise "[$A$]." An empty abstract forms a non-real number called a *black hole*. Nestings of abstract produce numerical values beyond the scope of this thesis and will not be discussed here.

Two axioms define the relationship between instance and abstract. The *involution* axiom defines them as functional inverses. When either boundary is immediately nested within the other, where nothing lies between them, both boundaries can be erased. The *distribution* axiom defines a further relationship between them: when an abstract boundary lies within an instance boundary, the instance boundary and abstract boundary and the content between them can be threaded across the contents of the abstract boundary. The visual forms in Section 4.2.2 make these rules visually apparent. The two axioms define the calculations necessary to reduce multiplication and power expressions to the canonical structure of natural numbers.

The two boundaries also construct a generalized form of *cardinality*. Cardinality rewrites a repeated reference using a single reference combined with a count of the repetition. In standard notation, a repeated sum is rewritten as a product with the count, e.g. $x + \ldots + x = nx$, and a repeated product is rewritten as an exponent of the count, e.g. $x \times \ldots \times x = x^n$. In the boundary calculus, cardinality takes the same form in addition and in multiplication. Because of this dual use, the boundary form of cardinality is considered to be a generalized form of cardinality.

This chapter begins by formally defining the two-boundary calculus.

## *4.2 The Two-Boundary Calculus*

The two-boundary calculus is introduced by defining the set of possible constructions, the *elements* of the calculus, and by defining *axioms* to state their equivalences.

### *4.2.1 Elements of the Calculus*

Elements of the calculus are constructed out of the void using two boundary distinctions: instance and abstract.

**Definition**. *Let* **B** *denote the collection of all well-formed boundary expressions composed of two boundaries, denoted as* (...) *and* [...]. **B** *is defined recursively by three rules:*

1. The void belongs to **B**.
2. If $a$ belongs to **B**, then ($a$) and [$a$] each belong to **B**.
3. If $b_1, b_2, \ldots, b_n$ $(n \geq 1)$ belong to **B**, then the unordered collection $b_1 b_2 \ldots b_n$ belongs to **B**.

The first rule establishes the void as the foundational element in the recursive construction of **B**. Thus, **B** contains the void.

The second rule creates elements of **B** by nesting other elements inside the two boundaries. The first two elements introduced by this rule are the boundaries with no content: ∘ and □. Further distinction introduces the following elements into **B**:

(∘),(□),[∘],[□],((∘)),((□)),([∘]),([□]),[(∘)],[(□)],[[∘]],[[□]].

The third rule creates new elements of **B** by *collecting* previous elements. These collections are unordered so permutations are irrelevant. This lack of order means that the form ○□ is considered identical to □○. Collection introduces the following elements into **B**:

○○, ○□, □□, ○(○), □(○), (○)(○), ○○○, ○○□, ○□□, □□□.

Between the three rules, deep and wide expressions are introduced into **B**, including the forms: ([○○○][○○]), (([[○○○]][○○])).

### *4.2.2 Equivalence Axioms*

Elements in the two-boundary calculus are related by two equivalence axioms: involution and distribution. The involution axiom defines a symmetric relationship between the two boundaries.

**Axiom 1 (Involution)** *Instance and abstract are functional inverses:*

$$([A]) \doteq A \doteq [(A)].$$

Involution allows the removal or introduction of instance-abstract pairs, including void equivalents when $A$ is void, (□)=[○]= . Involution does not apply to a pair with something lying between the boundaries; it can neither introduce or remove a configuration such as (○[○○○]), where ○ lies outside of the abstract and more than one instance lies within it. Acceptable examples of involution are shown in Figure 4.1.

The distribution axiom defines an asymmetric relationship between the two boundaries.

**Axiom 2 (Distribution)** *An instance around abstract distributes over the contents of the abstract:*

$$(A[BC]) \doteq (A[B])(A[C]).$$

Distribution manipulates the *modifier form*, $(A[\ldots])$, composed of instance, the template variable $A$, and abstract. It threads the modifier form over the contents of its inner abstract boundary. Distribution states that the form can modify these contents collectively or separately. Unlike involution, distribution matches a pattern lying between the two boundaries. Examples of distribution are shown in Figure 4.2.

These two axioms define the transformational basis of two-boundary expressions.

**Involution:** ([$A$])$\doteq A \doteq$[($A$)]

| **Equivalent Expressions** | **Template Replacement** |
|---|---|
| ([∘∘]) | |
|   ∘∘ | $A$ = ∘∘ |
| | |
| (   ([∘∘∘][∘∘])   ) | |
| (([([∘∘∘][∘∘])])) | $A$ = ([∘∘∘][∘∘]) |
| | |
| [   (∘)([(∘∘)   (∘∘∘)])] | |
| [   (∘)   (∘∘)   (∘∘∘)   ] | $A$ = (∘∘)(∘∘∘) |
| [([(∘)   (∘∘)])(∘∘∘)   ] | $A$ = (∘)(∘∘) |
| | |
| (   [∘][([∘∘]   [∘∘∘])]) | |
| (   [∘]   [∘∘]   [∘∘∘]   ) | $A$ = [∘∘][∘∘∘] |
| ([([∘]   [∘∘])][∘∘∘]   ) | $A$ = [∘][∘∘] |
| | |
| ([∘∘∘]      ) | |
| ([∘∘∘][∘]) | $A$ = |
| | |
| (([[(∘)]][∘∘∘])) | |
| (([   ∘   ][∘∘∘])) | $A$ = ∘ |
| ((         [∘∘∘])) | $A$ = |
| (            ∘∘∘   ) | $A$ = ∘∘∘ |

Figure 4.1: Examples of Involution.

**Distribution:** $(A[BC]) \doteq (A[B])(A[C])$

| Equivalent Expressions | Template Replacement |
|---|---|
| ([ooo][o])([ooo][<o>]) | |
| ([ooo][o <o>]) | $A$ = [ooo], $B$ = o, $C$ = <o> |
| | |
| (([[oooo]][])) | |
| (([[oooo]][])([[oooo]][])) | $A$ = [[oooo]], $B$ = , $C$ = |
| | |
| ([oo][ooo ooo]) | |
| ([oo][ooo])([oo][ooo]) | $A$ = [oo], $B$ = ooo, $C$ = ooo |
| ([oo oo][ooo]) | $A$ = [ooo], $B$ = oo, $C$ = oo |

Figure 4.2: Examples of Distribution.

### *4.3 Natural Numbers*

The elements of the two-boundary calculus map to natural numbers, complete with arithmetic operations of addition, multiplication, and exponentiation.

Space is treated as addition so that collecting elements adds them. Natural numbers are constructed simply by accumulating the unit formed by the empty instance boundary, $\circ$. The elements $\circ$, $\circ\circ$, $\circ\circ\circ$,... form the set of natural numbers, **N**, within the boundary calculus. The successor function, $S(A) \rightarrow A\circ$, inductively builds **N** from the element $\circ$.

#### *4.3.1 Addition*

Elements $a, b \in \mathbf{N}$ add by collection, $ab$. The set **N** is closed under addition because collecting elements can only form an element of equal or greater cardinality, which still an element of **N**

Addition is commutative because spatial collection is unordered. Addition is associative because spatial collection makes no grouping distinctions for multiple additions. The additive identity is the void: an element remains uneffected when collected with the void.

When adding these natural numbers, no calculation is necessary. They are initially phrased in their canonical form. For example, $3 + 2$ rewrites as ∘∘∘ ∘∘.

### *4.3.2 Multiplication*

Elements $a, b \in \mathbf{N}$ multiply by the form ([$a$][$b$]). The set $\mathbf{N}$ is closed under the multiplication.

**Proof.** The product of $a \in \mathbf{N}$ and initial element, ∘, reduces to $a$ by involution, ([$a$][∘])=([$a$])=$a$. If the product of $a$ and $b$, ([$a$][$b$]), is in the $\mathbf{N}$, the product of $a$ and the successor of $b$ is also in the set,

$$([a][b\circ])=([a][b])([a][\circ])=([a][b])a.$$

Because addition is closed, this result is in $\mathbf{N}$. Therefore, by induction all products of $a, b \in \mathbf{N}$ are in $\mathbf{N}$.

Multiplication is commutative because [$a$] and [$b$] are unordered within the outer instance. Associativity of binary multiplication is shown by two applications of involution:

$$([([a][b])][c])=([a][b][c])=([a][([b][c])]).$$

The multiplicative identity is the unit, ∘. An element multiplied by the unit reduces to itself by involution,

$$([a][\circ])=([a])=a.$$

In the boundary calculus, the void represents zero. Accordingly, multiplication by the void is void, ([$a$][])= .

**Proof.** By distribution, a product with the void is equal to two copies of the same,

$$([a][\ ])([a][\ ])=([a][\ ]).$$

The equation $ee = e$ is true when $e$ is void or when collection of $e$ is idempotent. Since collection is assumed additive, $ee \neq e$ unless $e$ is void. Therefore, ([$a$][ ])= .

Multiplication of natural numbers requires some calculation to return values to the canonical form of collections of units. This reduction uses distribution to strip away units one at a time, along with involution to remove unit multiplications.

| | |
|---|---|
| $3 \times 2$ | Given |
| ooo $\times$ 2 | Number Rewrite |
| ([ooo][oo]) | Multiplication Rewrite |
| ([ooo][o])([ooo][o]) | Distribution |
| ([ooo] )([ooo] ) | Involution |
| oooooo | Involution |

*4.4 Cardinality*

In the boundary calculus, repeated references to the same element can be rewritten as a cardinality of that element.

**Theorem 1 (Cardinality)** *Multiple references in the same context may be reduced to a single reference with multiple units signifying its quantity, as*

$$A...A\text{=([}A\text{][}\circ...\circ\text{]).}$$

Cardinality differs from multiplication in that this transformation is not semantically tied to its context, as it is in multiplication. Cardinality applies at any depth.

Everything has a count of one, shown directly by involution:

$$A\text{=([}A\text{])=([}A\text{][}\circ\text{]).}$$

Here, the lone $\circ$ denotes the single cardinality of $a$. Greater cardinalities can be formed by collecting many units in this space using distribution of separate single cardinalities. For example, two references can be distributed into a cardinality of two:

$$AA\text{=([}A\text{][}\circ\text{])([}A\text{][}\circ\text{])=([}A\text{][}\circ\circ\text{]).}$$

The induction step to all natural numbers is,

$$A...AA\text{=([}A\text{][}\circ...\circ\text{])([}A\text{][}\circ\text{])=([}A\text{][}\circ...\circ\circ\text{]).}$$

### *4.4.1 Dominion*

A zero cardinality, $([A]\square)$, reduces to the void as a void multiplication. In this form, the empty abstract, $\square$, dominates its context. This element is called a *black hole* because of this collapsing property. A general form of this collapsing derives from void multiplication.

**Theorem 2 (Dominion)** *Elements collected with a black hole are irrelevant, as*

$$\square A = \square.$$

Dominion is proven from void multiplication with an involution:

$$\square A = [(\square A)] = \square.$$

### *4.4.2 Generalized Cardinality*

The cardinality form is independent of its context and can be applied to repetition in both addition and multiplication. Cardinality applies to all repetition in the same context, regardless of the form repeated or the depth of this context. Any collection of identical elements can be counted this way.

Cardinality applies to addition directly. It applies to multiplication by counting abstracted elements inside the context of a surrounding instance boundary. For example, $([a][a]) = (([[a]][\circ\circ]))$. Cardinality of multiplication translates to exponentiation. Cardinality appears in standard and boundary numbers as:

$$x + \ldots + x = nx \quad \text{versus} \quad x \ldots x = ([x][\circ \ldots \circ])$$
$$x \times \ldots \times x = x^n \quad \text{versus} \quad ([x] \ldots [x]) = (([[x]][\circ \ldots \circ]))$$

## *4.5 Conclusions*

Though a single boundary represents natural numbers, a second boundary provides convenient forms for algebraic multiplication and exponentiation. The involution and distribution axioms completely define computation on the natural numbers for these functions. The two-boundary calculus is summarized in Table 4.1.

The two-boundary calculus constructs a generalized form of cardinality. This form applies to all situations of repeated reference in space, including addition and

Table 4.1: The Two-Boundary Calculus.

| *Numbers* | *Operators* | *Rules* |
|---|---|---|
| $0 \to$ | $x + y \to xy$ | $([A]) \doteq A \doteq [(A)]$ |
| $1 \to \circ$ | $x \times y \to ([x][y])$ | $(A[BC]) \doteq (A[B])(A[C])$ |
| $2 \to \circ\circ$ | $x^y \to (([[x]][y]))$ | |
| $3 \to \circ\circ\circ$ | | |

multiplication. It provides a form that is unavailable in standard notation. Though cardinality is an important concept, standard notation does not have a form for it that is independent of context. Cardinality provides a substrate for building a base system and thus a route to remove the representational inconvenience of the unary notation for numbers in Table 4.1.

The two-boundary calculus is similar to Spencer-Brown's numbers, except that it uses two boundaries where Spencer-Brown uses just one [10]. The additional boundary disambiguates the situations that his system had difficulty with. It still includes remnants of his logical arithmetic with involution and dominion. Ideally, this calculus would be defined as a numerical arithmetic, from rules such as involution and dominion, and generalized to algebraic axioms.

Chapter 5

# INVERSE

## 5.1 *Introduction*

This chapter extends boundary numbers to integers, rationals, and algebraic irrationals by introducing a third boundary to serve as a generalized inverse.

The first two boundaries, instance and abstract, remain exactly as defined in Chapter 4. The first two axioms, involution and distribution, are also the same.

The third boundary extends this system with a generalized inverse. This boundary shall be called the *inverse* boundary and be written as a triangle when empty, “$\triangle$,” or as angled brackets otherwise, “<*A*>.” An inverse boundary with no content is equivalent to void. The inverse boundary is its own functional inverse, so nesting inverse inside of inverse cancels out.

The characteristic property of the inverse boundary is defined by the *inversion* axiom. An element and its inverse cancel to the void. Similarly, an element and its inverse can be introduced from the void. All elements have an inverse except for the black hole.

Inverse serves directly as the additive inverse (i.e. $-x$) and, in a construction with instance and abstract, as the multiplicative inverse (i.e. $1/x$). Reduction of both inverses is handled by the inversion axiom, cancelling a form collected with its inverse. The context of this cancellation unambiguously determines the semantics as the subtraction (i.e. $x - x = 0$) or division (i.e. $x/x = 1$). Because the inverse boundary serves both addition and multiplication, it is considered to be a generalized inverse.

This chapter begins by redefining the calculus with three boundaries.

## 5.2 *The Three-Boundary Calculus*

The two-boundary calculus from Chapter 4 is extended with a third boundary by redefining the set of elements to include it and by adding a third axiom to define its transformations.

### *5.2.1 Elements of the Calculus*

Elements of the calculus are constructed out of the void using three boundary distinctions: instance, abstract, and inverse.

**Definition.** Let **B** denote the collection of all well-formed boundary expressions composed of three boundaries, denoted as `()`, `[]`, and `<>`. **B** is defined recursively by four rules:

1. The void belongs to **B**.

2. If $a$ belongs to **B**, then ($a$) and [$a$] each belong to **B**.

3. If $b$ belongs to **B** and $b \neq \square$, then <$b$> belongs to **B**.

4. If $c_1, c_2, \ldots, c_n$ $(n \geq 1)$ belong to **B**, then the unordered collection $c_1 c_2 \ldots c_n$ belongs to **B**.

Rules 1, 2 and 4 are identical to those in Chapter 4. The first rule establishes the void as the foundational element in **B**. The second rule creates elements by nesting elements in the instance and abstract boundaries. The fourth rule creates elements by collecting other elements.

The new, third rule creates elements by nesting other elements in the inverse boundary. It works identically to the second rule, except that the black hole is excluded. In other words, <□> is undefined. The first element introduced by this rule is the inverse boundary with no content: △. The distinction rules (2 and 3) introduce the following elements into **B** from the void:

(△),[△],<△>,<∘>,(<∘>),[<∘>],<(∘)>,<(□)>,<(<∘>)>,<[<∘>]>,<<(∘)>>.

Between the four rules, deep and wide expressions are introduced into **B**, including the forms: <(<[∘∘∘]>)>, (([[<∘>]]<[∘∘]>)).

### 5.2.2 Equivalence Axioms

Elements in the three-boundary calculus are related by three equivalence axioms: involution, distribution, and inversion. The involution and distribution axioms are defined as before.

**Axiom 1 (Involution)** *Instance and abstract are functional inverses:*

$$([A])\doteq A\doteq[(A)].$$

**Axiom 2 (Distribution)** *An instance around abstract distributes over the contents of the abstract:*

$$(A[BC])\doteq(A[B])(A[C]).$$

The new axiom of inversion defines transformations with the inverse boundary.

**Axiom 3 (Inversion)** *An element and its inverse are equivalent to void:*

$$A<A>\doteq$$

Inversion matches if the complete contents of an inverse boundary are found in the context of that boundary. Examples of inversion are given in Figure 5.1.

Inversion uses void substitution. An element and its inverse can be introduced anywhere in a boundary expression, so long as both elements are put into the context. Likewise, an element and its inverse can be removed from a boundary expression, so long as both reside in the same context.

These three axioms define all of the transformations of the calculus.

## 5.3 The Additive Inverse

The inverse boundary serves directly as the additive inverse. The additive inverse extends the natural numbers built in Chapter 4 to the integers.

Adding an element to its additive inverse reduces to the additive identity, the void. For example, $3-3=0$ translates to $\circ\circ\circ<\circ\circ\circ> =$ . The additive inverse appears generally in standard and boundary numbers as:

$$x+(-x)=0 \quad \text{versus} \quad x<x>= \ .$$

**Inversion:** $A<A>\doteq$

| **Equivalent Expressions** | **Template Replacement** |
|---|---|
| oooo<oo> | |
| oo | $A$ = oo |
| | |
| ([ooo]) | |
| ([ooo][oo]<[oo]>) | $A$ = [oo] |
| | |
| <<(ooo)>> | |
| <<(ooo)>><(ooo)>(ooo) | $A$ = (ooo) |
| (ooo) | $A$ = <(ooo)> |

Figure 5.1: Examples of Inversion.

### *5.3.1 Properties of the Additive Inverse*

The inverse boundary can be manipulated in ways analogous the the additive inverse. The basic properties of the inverse are outlined by three theorems: inverse collection, inverse cancellation, and inverse promotion. The theorem of inverse collection transforms collected inversions into a single inversion and the theorem of inverse cancellation removes pairs of nested inverses.

**Theorem 1 (Inverse Collection)** *A collection of inverted elements equals an inversion of the collection, as*

$$<A><B>=<AB>.$$

A proof of inverse collection follows from inversion:

$$<A><B>=<A><B><AB>AB=<AB>.$$

**Theorem 2 (Inverse Cancellation)** *The inverse boundary is its own functional inverse, as*

$$<<A>>=A.$$

A proof of inverse cancellation also follows from inversion:

$$<<A>>=<<A>><A>A=A.$$

Besides collecting and cancelling, the inverse operation can be "promoted" over the modifier form, $(A[\ldots])$. The theorem of inverse promotion derives this property.

**Theorem 3 (Inverse Promotion)** *An inverse boundary can be promoted over the composite boundary,* $(A[\ldots])$*:*

$$<(A[B])>=(A[<B>]).$$

**Proof.** For all $A, B, C \in \mathbf{B}, B \neq \square$,

| | |
|---|---|
| $<(A[B])>$ | Given |
| $<(A[B])>(A[\quad\quad])$ | Void Multiplication |
| $<(A[B])>(A[B<B>])$ | Inversion |
| $<(A[B])>(A[B])(A[<B>])$ | Distribution |
| $(A[<B>])$ | Inverse |

These three theorems appear in standard and boundary numbers as:

$$(-x)+(-y) = -(x+y) \quad \text{versus} \quad <x><y>=<xy>$$
$$-(-x) = x \quad \text{versus} \quad <<x>>=x$$
$$-(xy) = x(-y) \quad \text{versus} \quad <(x[y])>=(x[<y>])$$

### *5.3.2 Integers*

Since the boundary calculus now includes the additive inverse, natural numbers can be extended to the integers. The instance boundary serves as a unit for the natural numbers and the inverse boundary inverts that unit. The negative unit is given by $<\circ>$. These numbers add by spatial collection and multiply by the form using the abstract boundary.

The void and elements $\circ, <\circ>, \circ\circ, <\circ\circ>, \circ\circ\circ, <\circ\circ\circ>, \ldots$ compose the set $\mathbf{I}$ of integers within the boundary calculus. The successor function is $S(A) \rightarrow A\circ$ and the predecessor function is $P(A) \rightarrow A<\circ>$. From the void, successors to the void take the form $\circ\ldots\circ$ and predecessors to the void take the form $<\circ\ldots\circ>$. The predecessor function gives $<\circ\ldots\circ><\circ>$, which equals $<\circ\ldots\circ\circ>$ by inverse collection.

Boundary integers add by collection, as $ab$, and are closed under addition.

**Proof.** Elements $a, b \in \mathbf{I}$ assume one of three forms: the void, a positive integer, or a negative integer.

1. If either is the void, the sum reduces to the other value, which is in **I**.
2. If both are positive integers, the sum is comprised of all units, which is in **I**.
3. If both are negative integers, the sum is of the form <o. . .o><o. . .o> which equals <o. . .oo. . .o> by inverse collection. This form is in **I**.
4. If $a$ and $b$ are of opposite sign and the positive number has fewer units, the negative number can be split by inverse collection and that part cancelled out, e.g. o. . .o<o. . .oo. . .o>=o. . .o<o. . .o><o. . .o>=<o. . .o>. This result is in **I**.
5. Otherwise, if $a$ and $b$ must be of opposite sign with the positive number having more units. In this case, the negative number cancels directly with part of the positive value, e.g. o. . .oo. . .o<o. . .o>=o. . .o. This result is in **I**.

Addition is commutative because spatial collection is unordered. Addition is associative because spatial collection makes no grouping distinctions for multiple additions. The additive identity is still void.

Every element in **I** has an additive inverse.

**Proof.** The inverse of the void is the void, <>= , derived directly by inversion. All successors to the void, the natural numbers of the form o. . .o, have as additive inverses the same value wrapped by inverse, <o. . .o>. All predecessors to the void, the natural numbers wrapped by the inverse boundary as <o. . .o>, have as additive inverses the same value wrapped again by the inverse <<o. . .o>>, which reduces to the uninverted natural number of the form o. . .o by inverse cancellation.

Integers $a, b \in \mathbf{I}$ multiply with the same form used for natural numbers, ([$a$][$b$]). The integers are closed under multiplication.

**Proof.** Elements $a, b \in \mathbf{I}$ each assume one of three forms: the void, a positive integer, or a negative integer. Let $c, d \in \mathbf{N}$ be the magnitude of $a, b$ respectively.

1. When either is the void, the product reduces to the void by void multiplication, which is in **I**.

2. When both are positive, the product is in **I** because natural numbers are closed under multiplication.

3. When one of them is negative, as ([*c*][<*d*>]), the product is the inverse of the product of their magnitudes, <([*c*][*d*])>, by inverse promotion. Since the product of two natural numbers is a natural number and the inverse of a natural number is an integer, this result is an integer.

4. When both numbers are negative, as ([<*c*>][<*d*>]), the product is the inverse-inverse of the product of their magnitudes, <<([*c*][*d*])>>, by inverse promotion applied twice. This reduces to ([*c*][*d*]), the product of two natural numbers, which is a natural number.

Multiplication is commutative because [*a*] and [*b*] are unordered within the outer instance. Multiplication is associative by two applications of involution:

([([*a*][*b*])][*c*])=([*a*][*b*][*c*])=([*a*][([*b*][*c*])]).

### *5.3.3 Calculation*

Calculation with the additive inverse is straightforward, seeking to eliminate abstract and multiple inverses. Subtraction requires matching of positive and negative quantities:

| | |
|---|---|
| $3 - 5$ | Given |
| ooo − ooooo | Number Rewrite |
| ooo<ooooo> | Subtraction Rewrite |
| ooo<ooo><oo> | Inverse Collection |
| <oo> | Inversion |

Multiplication requires management of the inverse.

| | |
|---|---|
| $-3 \times -2$ | Given |
| `<ooo>  ×  <oo>` | Number Rewrite |
| `([<ooo>][<oo>])` | Multiplication Rewrite |
| `<([<ooo>][ oo ])>` | Inverse Promotion |
| `<<([ ooo ][ oo ])>>` | Inverse Promotion |
| `([ ooo ][ oo ])` | Inverse Cancellation |
| `([ooo][o])([ooo][o])` | Distribution |
| `ooo       ooo` | Cardinality |

### 5.4 *Multiplicative Inverse*

The inverse serves as the multiplicative inverse when set between the instance and abstract boundaries, as `(<[a]>)`.

Multiplying something against its additive inverse reduces to the multiplicative identity, the unit. For example, $3/3 = 1$ translates to `([ooo]<[ooo]>)=()`. The multiplicative inverse appears in standard and boundary numbers as:

$x/x = 1$ versus `([x]<[x]>)=()`

#### 5.4.1 *Properties of the Multiplicative Inverse*

The theorems for the additive inverse also apply to the multiplicative inverse. For simplicity in this discussion, some involution steps are assumed. For example, when multiplying by a multiplicative inverse, involution immediately applies:

`([a][(<[b]>)])=([a]<[b]>)`.

Inverse collection states that the product of two inverted numbers equals an inversion of their product. Inverse cancellation states that multiplicative inverse cancels itself out. These two theorems for multiplication appear in standard and boundary numbers as:

$1/x \times 1/y = 1/xy$ versus `(<[x]><[y]>)=(<[x][y]>)`

$1/(1/x) = x$ versus `(<[(<[x]>)]>)=x`.

Inverse promotion carries the inverse over the modifier form, as $<(a[b])>=(a[<b>])$. This theorem translates to standard numbers as the conversion between the multiplicative inverse of a value and the additive inverse of its exponent. Inverse promotion for multiplication appears in standard and boundary numbers as:

$$1/x^y = x^{-y} \quad \text{versus} \quad (<([[x]][y])>)=(([[x]][<y>]))$$

### *5.4.2 Rationals*

Because the boundary calculus now includes the multiplicative inverse, integers can be extended to rationals. The multiplicative inverse of $a$ is simply $(<[a]>)$. For $n, d \in \mathbf{I}$ with $d \neq$ , the element $([n]<[d]>)$ is in $\mathbf{Q}$, the set of rationals.

The rationals, $a, b \in \mathbf{Q}$ add by collection $ab$ and are closed under addition.

**Proof.** Rational numbers $a, b \in \mathbf{Q}$ take the form $a=([i]<[j]>)$ and $b=([k]<[l]>)$, where $i, k, j, l \in \mathbf{I}$ with $j, l \neq$ . The sum of these rationals is $ab=([i]<[j]>)([k]<[l]>)$, which reduces to rational $c=([m]<[n]>)$, where $m=([i][l])([k][j])$ and $n=([j][l])$, by the following derivation.

| | |
|---|---|
| $(\ [i]\ \ \ \ \ \ <[j]>)(\ [k]\ \ \ \ \ \ <[l]\ >)$ | Given |
| $(\ [i][l]\ <[l]><[j]>)(\ [k][j]\ <\ [j]><[l]\ >)$ | Inversion |
| $(\ [i][l]\ <[l]\ [j]>)(\ [k][j]\ <\ [j]\ [l]\ >)$ | Inverse Collection |
| $([([i][l])]<[l]\ [j]>)([([k][j])]<[([j]\ [l])]>)$ | Involution |
| $([([i][l])\ ([k][j])]<[([j]\ [l])]>)$ | Distribution |
| $([\ m\ ]<[\ n\ ]>)$ | Rewrite |

Since the product of two integers in $\mathbf{I}$ is an integer, $n$ is an integer. Because $j$ and $l$ are non-void, their product is non-void. Products $([i][l])$ and $([k][j])$ are also integers. Since the sum of two integers is an integer, $m=([i][l])([k][j])$ is also an integer. Therefore, $c$ is rational, $c \in \mathbf{Q}$, and addition is closed under the rationals.

Every element in $\mathbf{Q}$ has an additive inverse, found simply by wrapping the inverse boundary around it.

Rationals, $a, b \in \mathbf{Q}$ multiply by the same form used for natural numbers, $([a][b])$. The rationals are closed under multiplication.

**Proof.** Rational numbers $a, b \in \mathbf{Q}$ take the form $a=([i]<[j]>)$ and $b=([k]<[l]>)$, where $i, k, j, l \in \mathbf{I}$ with $j, l \neq$ . The product of these rationals is also in $\mathbf{Q}$ because the product of $a$ and $b$ reduces to the rational $([m]<[n]>)$, where $m=([i][k])$ and $n=([j][l])$.

| | |
|---|---|
| $([\quad a \quad][\quad b \quad])$ | Given |
| $([([i]<[j]>)][([k]<[l]>)])$ | Replacement |
| $(\quad [i]<[j]> \quad [k]<[l]> \quad)$ | Involution |
| $(\quad [i]\ [k] \quad <[j]\ [l]> \quad)$ | Inverse collection |
| $([([i]\ [k])]<[([j]\ [l])]>)$ | Involution |
| $([\quad m \quad]<[\quad n \quad]>)$ | Replacement |

Since the product of two integers is an integer, both $m$ and $n$ are integers. Because $j$ and $l$ are non-void, $n$ is also non-void. Therefore, $([m]<[n]>)$ is rational and so the product of $a$ and $b$ is rational.

Every element in $\mathbf{Q}$, except for the void, has a multiplicative inverse.

**Proof.** Rational number $a \in \mathbf{Q}$ takes the form $a=([i]<[j]>)$, where $i, k \in \mathbf{I}$ with $j \neq$ . Since $a$ is non-void, the numerator is also non-void, $i \neq$ . The multiplicative inverse of $a$, given by $(<[a]>)$, is rational by the following reduction back to the rational form.

$$(<[a]>)=(<[([i]<[j]>)]>)=(<[i]<[j]>>)=(<[i]>[j])$$

### *5.4.3 Division by Zero*

Dominion, $a\square=\square$, loses information and necessitates excluding $<\square>$ from the calculus. This restriction prevents formation of division by zero, $(<\square>)$. This restriction appears in standard and boundary numbers as:

$$1/0 \textit{ undefined} \quad \text{versus} \quad <\square> \textit{ undefined.}$$

### *5.4.4 Calculation*

Computing a division requires a matching of quantities. When quantities cannot be matched, the fraction does not reduce. Thus, the calculus constrains this calculation but does not guide the reduction.

| | |
|---|---|
| 6/2 | Given |
| ([oooooo]<[oo]>) | Rewrite |
| ([([ooo][oo])]<[oo]>) | Cardinality |
| (   [ooo][oo]   <[oo]>) | Involution |
| (   [ooo]            ) | Inversion |
| ooo | Involution |

### 5.5 *Inverse and Cardinality*

The inverse applied to the cardinality form gives negative and fractional cardinalities.

#### 5.5.1 *Negative Cardinality*

A negative cardinality is one which cancels out with a positive cardinality of equal magnitude. A negative cardinality is indicated by the inverse boundary around the units. For instance, a cardinality of negative two cancels with a cardinality of two.

([*a*][oo])([*a*][<oo>])=([*a*][oo<oo>])=([*a*][])=

An inverted multiplicative form is changed to a negative cardinality by the theorem of inverse promotion.

<([*a*][o...o])>=([*a*][<o...o>])

For addition, this negative cardinality translates to multiplication by a negative integer. For multiplication, this negative cardinality translates to exponentiation by a negative integer. These forms appear in standard and boundary numbers as:

$-(nx) = (-n)x$ versus <([*x*][o...o])>=([*x*][<o...o>])

$1/x^n = x^{-n}$ versus (<([[*x*]][o...o])>)=(([[*x*]][<o...o>]))

#### 5.5.2 *Fractional Cardinality*

The multiplicative inverse differs from the additive inverse by placing the inverse boundary outside of an abstract boundary, instead of inside of it. Doing the same to the cardinality construction makes fractional cardinalities.

For example, a half-cardinality is given by ([*a*]<[oo]>). Collection of two of these reduces to $a$:

| | |
|---|---|
| ([*a*]<[oo]>)([*a*]<[oo]>) | Given |
| ([([*a*]<[oo]>)][oo]) | Cardinality |
| ([*a*]<[oo]>[oo]) | Involution |
| ([*a*]) | Inversion |
| $a$ | Involution |

Applied to addition combinations, fractional cardinality builds fractions. The form ([*a*]<[oo]>) translates to $a/2$. In the above proof this fraction is doubled, as

$$a/2 + a/2 = 2(a/2) = a(2/2) = a.$$

Applied to multiplicative combinations, fractional cardinalities build roots. The form (([[*b*]]<[oo]>)) translates to $b^{1/2}$. In a proof similar to the above but with $a$=[*b*], this root is squared to double the cardinality, as

$$b^{1/2}b^{1/2} = (b^{1/2})^2 = b^{2/2} = b.$$

Since fractional cardinalities form roots, they allow construction of the algebraic irrationals.

## *5.6 Conclusions*

Though the two-boundary calculus only represents natural numbers, simply adding an inverse boundary extends the boundary numbers through the algebraic irrationals. The three axioms completely define computation on these numbers and functions. The three-boundary calculus is summarized in Table 5.1.

The inverse boundary serves many inverse functions due to the structures provided by the instance and abstract boundaries. It is defined in the most fundamental way that an inverse can be defined, utilizing the void to cancel inverses to nothing. The spatial formalism makes this inverse definition possible and powerful. Kauffman defined an inverse boundary of this sort but did not have the other structures to utilize it in a general form.

The inverse is powerful but it introduces problems, such as division by zero. Though the two-boundary calculus was solid, the three-boundary calculus has a single exception, preventing rules from applying to all configurations. When performing

Table 5.1: The Three-Boundary Calculus

*Numbers*

$0 \rightarrow$

$1 \rightarrow \circ$

$2 \rightarrow \circ\circ$

$-1 \rightarrow \texttt{<}\circ\texttt{>}$

$-2 \rightarrow \texttt{<}\circ\circ\texttt{>}$

$1/2 \rightarrow \texttt{(<[}\circ\circ\texttt{]>)}$

$2/3 \rightarrow \texttt{([}\circ\circ\texttt{]<[}\circ\circ\circ\texttt{]>)}$

$\sqrt{3} \rightarrow \texttt{(([[}\circ\circ\circ\texttt{]]<[}\circ\circ\texttt{]>))}$

$\sqrt[3]{2} \rightarrow \texttt{(([[}\circ\circ\texttt{]]<[}\circ\circ\circ\texttt{]>))}$

*Operators*

$x \rightarrow x$

$-x \rightarrow \texttt{<}x\texttt{>}$

$1/x \rightarrow \texttt{(<[}x\texttt{]>)}$

$x + y \rightarrow xy$

$x - y \rightarrow x\texttt{<}y\texttt{>}$

$x * y \rightarrow \texttt{([}x\texttt{][}y\texttt{])}$

$x/y \rightarrow \texttt{([}x\texttt{]<[}y\texttt{]>)}$

$x^y \rightarrow \texttt{(([[}x\texttt{]][}y\texttt{]))}$

$x^{-y} \rightarrow \texttt{(([[}x\texttt{]][<}y\texttt{>]))}$

$\sqrt[y]{x} \rightarrow \texttt{(([[}x\texttt{]]<[}y\texttt{]>))}$

*Rules*

$$\texttt{([}A\texttt{])} \doteq A \doteq \texttt{[(}A\texttt{)]}$$

$$\texttt{(}A\texttt{[}BC\texttt{])} \doteq \texttt{(}A\texttt{[}B\texttt{])(}A\texttt{[}C\texttt{])}$$

$$A\texttt{<}A\texttt{>} \doteq$$

a computation, constraints on variables must be propagated through to avoid paradoxes.

# Chapter 6

# PHASE

## *6.1 Introduction*

This chapter extends the boundary calculus to complex numbers and basic transcendentals by interpreting boundaries functionally and by adding some manipulative constraints.

The inverse boundary can be put into an arithmetic form to which cardinality can be applied (the boundary itself cannot be counted). This form is called the *phase element*, or $J$, and it translates to the radian value $i\pi$ in standard numbers. Fractional cardinalities of $J$ build into the complex roots of one, just as fractions of $i\pi$ determine a radian angle in the complex plane. This property of J undermines additive space because it has indeterminate sign and ambiguous magnitude, a complication similar to using radian numbers to represent complex numbers. This issue is addressed in terms of cardinality and inverse.

The instance and abstract boundaries act as exponential and logarithmic functions, respectively. Interpreting these as the natural exponent, $e^x \rightarrow (x)$, and the natural logarithm, $\ln x \rightarrow [x]$, introduces these transcendental functions into the calculus. With these functions, basic transcendentals such as $e$ and $\pi$ can be constructed.

## *6.2 Phase*

When manipulating boundary expressions, the inverse boundary falls into an arithmetical form that cardinality can apply to.

Recall the concept of negative cardinality, $<a>=([a][<\circ>])$, written as $-a = a \times -1$ in standard notation. In the boundary form, the inverse boundary has been separated from what it previously contained, into the encapsulated form $[<\circ>]$. This form shall be called $J$:

$$J \doteq [<\circ>].$$

*J* has a stability unparalleled in the calculus: it uses all three boundaries exactly once. All other legal configurations of three different boundaries reduce to void because the abstract and instance boundaries cancel out. (The sixth case, (<[]>), is undefined.)

[(<>)],([<>]),<[()]>,<([])>.

### *6.2.1 Phase Independence*

The configuration of three boundaries also serves as *phase operator*, given as [<(*A*)>]. This operator possesses a curious property of independence from its contents.

**Theorem 1 (Phase Independence)** *In a nesting of abstract, inverse, and instance boundaries, the contents of the instance can be moved to the context of the configuration.*

**Proof.** For all $A \in \mathbf{B}$,

```
[<(A)>]                         Given
[<(A)> (   [             ])]    Involution
[<(A)> (A  [             ])]    Dominion
[<(A)> (A  [()      <()>])]     Inversion
[<(A)> (A  [()])(A  [<()>])]    Distribution
[<(A)> (A      )(A  [<()>])]    Involution
[                (A  [<()>])]   Inversion
                  A  [<()>]     Involution
```

Phase independence simplifies manipulations with the inverse. In particular, it simplifies the derivation of negative cardinality, as

<*A*>=([<([*A*])>])=([*A*][<o>]).

Phase independence also gives a different proof of inverse promotion. Rather than generate, distribute, and cancel (as in the proof in Section 5.3.1), with phase independence, the inverse is wrapped into the phase operator and moved.

<(*A*[*B*])>=([<(*A*[*B*])>])=(*A*[<([*B*])>])=(*A*[<*B*>]).

Using $J$, two concise and useful forms of inverse promotion become possible: the inverse jumping inside of the instance boundary and the inverse jumping outside of the abstract boundary:

$$<(A)>=([<(A)>])=(A[<()>])=(AJ),$$
$$[<A>]=[<([A])>]=[A][<()>]=[A]J.$$

*6.2.2 Oscillation*

The phase element has the property of being its own inverse. Only zero exhibits this property in rational numbers.

**Theorem 2 (J Cancellation)** *J cancels with itself.*

$$[<\circ>][<\circ>]=[<([<\circ>])>]=[<<\circ>>]=[\circ]=$$

That $J$ is its own inverse should be of no surprise, since the inverse boundary is its own functional inverse and $J$ embodies the inverse.

In light of this, $J$ can be built into an oscillation function, $osc(A) \rightarrow AJ$. Every second application causes the Js to cancel out, producing the sequence:

$$\rightarrow J \rightarrow \quad \rightarrow J \rightarrow \quad \rightarrow J \rightarrow \ldots$$

Wrapping the elements of this sequence with the instance boundary produces a more familiar sequence. Instead of starting with void, this sequence starts with $\circ$ and uses the oscillation function $osc(A) \rightarrow ([A]J)$, producing this sequence:

$$\circ \rightarrow (J) \rightarrow \circ \rightarrow (J) \rightarrow \circ \rightarrow (J) \rightarrow \ldots$$

Since $(J)=<\circ>$, the function is just a multiplication by negative one. The sequence translates to:

$$1 \rightarrow -1 \rightarrow 1 \rightarrow -1 \rightarrow 1 \rightarrow -1 \rightarrow \ldots$$

### *6.2.3 Cardinality of J*

The periodicity of this oscillation can be changed by applying cardinality to J. For example, the half-cardinality of $J$ produces a four-step oscillation with the function $osc(A) \rightarrow A([J]<[\circ\circ]>)$:

$$\rightarrow([J]<[\circ\circ]>)\rightarrow J\rightarrow J([J]<[\circ\circ]>)\rightarrow \quad \rightarrow([J]<[\circ\circ]>)\rightarrow\ldots$$

Wrapping the elements of this sequence with the instance boundary again produces a more familiar sequence. The oscillatory function this time is $osc(A) \rightarrow ([A]([J]<[\circ\circ]>))$:

$$\circ\rightarrow(([J]<[\circ\circ]>))\rightarrow(J)\rightarrow(J([J]<[\circ\circ]>))\rightarrow\circ\rightarrow(([J]<[\circ\circ]>))\rightarrow\ldots$$

Interpreting $(([J]<[\circ\circ]>))$ as $i$ reveals the oscillation function to be a multiplication by i, producing oscillation between 1 and $-1$, this time in four-steps:

$$1\rightarrow i\rightarrow -1\rightarrow -i\rightarrow 1\rightarrow i\rightarrow\ldots$$

The oscillation function, $osc(A) \rightarrow ([A]([J]<[\circ\circ]>))$, is a half-inverse operation. Similarly, other cardinalities of J can be taken to produce other periodicities, forming the roots of unity.

## *6.3 Multiple Value*

A half-inverse operation would be a powerful addition to the boundary calculus, except that cardinality of J is syntactically unstable. In particular, J has ambiguous sign and ambiguous (multi-valued) cardinality.

$$\begin{array}{ll} \textit{Ambiguous sign} & J=<J> \\ \textit{Ambiguous cardinality} & =JJ=JJJJ=JJJJJJJJ=\ldots \end{array}$$

The second property follows directly from J-cancellation. The first property follows from inversion and J-cancellation.

$$J=J<>=J<JJ>=J<J><J>=<J>.$$

The ambiguous sign makes the cardinalities of J have ambiguous signs also, disrupting the oscillation points. For example,

| | |
|---|---|
| `([ J ]<[oo]>)` | Given |
| `([<J>]<[oo]>)` | Ambiguous Sign |
| `<([ J ]<[oo]>)>` | Inverse Promotion |

The ambiguous cardinality of J occurs at all values.

| | |
|---|---|
| `([ J ]<[oo]>)` | Given |
| `([JJJ]<[oo]>)` | J Cancellation |
| `([ J ]<[oo]>)([JJ]<[oo]>)` | Distribution |
| `([ J ]<[oo]>)([J][oo]<[oo]>)` | Cardinality |
| `([ J ]<[oo]>)([J])` | Inversion |
| `([ J ]<[oo]>)J` | Involution |

J undermines the calculus because of these ambiguities. Cardinalities of J are useful as they provide complex numbers. Resolving this ambiguity restores the integrity of the calculus and extends it to complex numbers. Two steps accomplish this:

1. *Convert inverse boundaries to Js.* Whenever the inverse boundary is wrapped by the abstract boundary, as `[<A>]`, pull the inverse out, as `[A]J`.

2. *Limit J cancellation.* Allow Js to cancel only within an instance boundary and *after* applying other reductions.

These constraints are not exhaustive but they do approximate the constraints on collapsing complex numbers to radian values within complex exponentials.

### *6.4 Exponentials and Logarithms*

The instance and abstract boundaries can be functionally interpreted as an exponential operation and a logarithm operation, respectively. The base of these operations is not constrained by the calculus. Though it is not mandatory to specify, some choices are nevertheless convenient. For example, using base two or base ten directly provides logarithms and exponents to that magnitude.

Obviously the "natural" choice is to use base $e$, creating the interpretations $e^x \to$ `(x)` and $\ln x \to$ `[x]`. With these functions, the calculus can construct basic transcendental values and the trigonometry functions.

Multiplication in the calculus can be seen as an adding of logarithms:

$$xy = e^{\ln(xy)} = e^{\ln x + \ln y} \rightarrow \texttt{([}x\texttt{][}y\texttt{])}$$

Exponentiation in the calculus can be seen as a multiplication of a logarithm:

$$x^y = e^{\ln x^y} = e^{y \ln x} = e^{e^{\ln(y \ln x)}} = e^{e^{\ln y + \ln \ln x}} \rightarrow \texttt{(([[}x\texttt{]][}y\texttt{]))}$$

The logarithmic formulas connect multiplicative operations with additive operations. Because the forms of calculus can be interpreted as logarithms, these transformations are trivial.

$$\begin{aligned} \ln(xy) = \ln x + \ln y &\quad \text{versus} \quad \texttt{[([}x\texttt{][}y\texttt{])]=[}x\texttt{][}y\texttt{]} \\ \ln x/y = \ln x - \ln y &\quad \text{versus} \quad \texttt{[([}x\texttt{]<[}y\texttt{]>)]=[}x\texttt{]<[}y\texttt{]>} \\ \ln x^y = y \ln x &\quad \text{versus} \quad \texttt{[(([[}x\texttt{]][}y\texttt{]))]=([[}x\texttt{]][}y\texttt{])} \end{aligned}$$

## 6.5 *Transcendentals*

Transcendental numbers, such as Euler's number and an algebraic $\pi$, can be represented in the calculus as can basic trigonometric functions.

### 6.5.1 *Euler's Number, e*

The instance boundary acts as an exponential. A double nesting of instance equals its base, $e^{e^0} = e \rightarrow \texttt{(}\circ\texttt{)}$. Setting the base of the function to Euler's number makes this boundary form represent Euler's number. This nesting does not reduce to any other boundary form, making the transcendental value incommensurable with previously defined numbers.

### 6.5.2 *Pi, π*

Interpreting the abstract boundary as the natural logarithm makes $J = \ln -1 = i\pi$. This interpretation fulfills Euler's formula translated to J wrapped by instance:

$$1 + e^{i\pi} = 0 \quad \text{versus} \quad \circ\texttt{([<}\circ\texttt{>])=}$$

Table 6.1: Transcendentals in the Calculus.

| **Transcendental** | **Standard** | **Boundary** |
|---|---|---|
| $J$ | $i\pi$ | $[<\circ>]$ |
| $i$ | $\sqrt{-1}$ | $(([J]<[\circ\circ]>))$ |
| $\pi$ | $3.1415\ldots$ | $(J[J]([J]<[\circ\circ]>))$ |
| $e$ | $2.7183\ldots$ | $(\circ)$ |

From J and $i$, a radian interpretation of $\pi$ can be algebraically constructed.

$$\begin{aligned}\pi &= -1 \times i \times i\pi \\ &= ([<\circ>][(([J]<[\circ\circ]>))][J]) \\ &= (J[J]([J]<[\circ\circ]>))\end{aligned}$$

This construction of $\pi$ treats it not as a real number but as the radian unit. Its semantic value comes out of this algebraic construction and has no relation to its numerical value of $\pi = 3.14159\ldots$.

The numerical values of $e$ and $\pi$ in standard mathematics are based on criteria independent of the manipulations used here (e.g. geometry) and are not essential to the algebraic behavior of the transcendentals. In the calculus, they are devoid of numerical value and remain incommensurable with other quantities. The boundary representations of these basic transcendental values are given in Table 6.1.

### *6.5.3 Trigonometric Functions*

In the calculus, trigonometric functions can be computed symbolically using the three boundary rules. The function for cosine in standard and boundary forms appears as:

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}$$

versus

$$\cos x = ([(([x]([J]<[\circ\circ]>)))(([<x>]([J]<[\circ\circ]>)))]<[\circ\circ]>)$$

The cosine function behaves as expected, reducing to real results. For instance, $\cos \pi = -1$. Although the derivation can be lengthy and complicated, it is based entirely on the three axioms and the theorems that follow from them. Accordingly, trigonometric formulas can be derived from these axioms.

### *6.6 Conclusions*

The three-boundary calculus supports forms which act algebraically as complex numbers and basic transcendentals.

Surprisingly, the simple forms of the calculus extend to the more advanced mathematics of trigonometry and complex exponentials, which demonstrate the basic concept of manipulating the inverse.

Some simple observations of transcendental numbers arise in this discussion. The phase behavior of complex exponentials is in no way tied to the numeric values of $e$ and $\pi$. Assigning them numeric values serves altogether different purposes (e.g. derivatives).

Undefined and uninterpreted, the base of instance and abstract is transcendental because (o) is incommensurable with any other boundary construction. An equality could be introduced that defined this to be non-transcendental, such as base two with the definition (o) $\doteq$ oo. Everything in the calculus would still hold as none of it is dependent on this value.

Chapter 7

# FUTURE WORK

## *7.1 Introduction*

The ability to form and manipulate numerical expressions is a small part of number mathematics and numbers are just one domain of mathematics. The minimalist techniques of boundary mathematics can be applied to the larger context of numbers and to additional areas of mathematics. The representational paradigm may ultimately encompass much of mathematics.

Here, I consider the future of the calculus of number. The advancement of the calculus may proceed in three directions: towards wider coverage, that boundary numbers may be well-defined within the structures known to elementary algebra; towards practicality, that techniques for learning and working with math can be rediscovered with this new conceptualization; and towards extended uses of numbers, that integral calculus and other transformations may be reconceptualized under boundary mathematics. If they are sufficiently expanded boundary numbers may find practical use.

## *7.2 Coverage*

The mathematics of number is extensive and detailed. This calculus of number covers only part of it, showing that algebraic expressions and arithmetic computation can be done with boundaries. The calculus remains without many structures that would be required to do number mathematics.

*Arithmetic definition.* The given definition of the calculus is not an ideal one because it presents an algebraic definition. A proper definition would derive the algebraic axioms by generalizing from arithmetic laws on the basic forms. Choices of axioms need to be assessed and compared for their utility in describing the dynamics of the forms, working towards arithmetic laws.

*Resolve paradoxes.* This definition is not thorough because the paradoxes of multiple-value and zero singularity are not completely resolved. The traditional notation addresses these paradoxes by considering the domain and range of each function and then limiting rule application accordingly. This approach does not clearly translate to the boundary forms because the approach relies heavily on a fairly wide context of forms (i.e. numerical class) to determine when a rule applies. In the boundary calculus, the paradoxes must be considered at the level of boundaries, with as little dependence on number type as possible. A preliminary attempt at this was made in Chapter 6.

*Additional functions.* The boundary calculus can express only the basic arithmetic functions. It leaves out many useful functions found in elementary algebra. While it includes addition, subtraction, multiplication, division, exponents, and logarithms, those functions which utilize set theoretics, such as summations, or those which have discontinuities, such as absolute value, are not covered.

*Structures around expressions.* It lacks many structures for doing ordinary algebra, such as equality and inequality (standard linear structures were used here). It has no larger system of truth maintenance, nor a form for abstracting functions. Such basic tenets of mathematics are sorely necessary for any notation to be remotely useful. Since most of predicate calculus is well understood, this extension is reasonable.

In this discourse, these issues were avoided by embedding the boundary forms within traditional linear constructions. Because boundary calculus is spatial, many of its advantages are lost when it is restricted to linear constructs. To maintain the spatial advantages, the calculus most cover this completely, so that the entire notation is spatial.

## *7.3 Practical*

Currently, the boundary calculus is impractical for manipulating numbers, because traditional support tools and techniques do not apply directly to its forms. The common knowledge taught in grade school mathematics is directed towards standard notation and must be reconsidered to use with boundary numbers (see [29]).

*Problem phrasing.* Traditional *problem representation* techniques were designed around the conceptual objects of standard notation: numbers and functions. In contrast, the boundary forms use distinction and collection. It is unlikely that mathematics problems could be phrased directly in terms of the boundary constructs, though higher compositions of them, such as numbers and functions, certainly would work. If boundary forms are in any way fundamental, then thinking in terms of boundaries should provide a conceptual advantage over traditional forms.

*Algorithms.* Algorithms for carrying through calculations and finding solutions must be reinterpreted in this form. Even common arithmetic routines, such as adding based integers, requires a heuristic to direct the result to the proper form. In many cases, common heuristics will emerge that are relatively hidden in the traditional notation. Because the details of mathematical manipulation differ so much in boundary calculus, the high level manipulations may fall into remarkably different patterns.

*Management of forms.* The boundary calculus forces manipulation at the very lowest level. Constructs that are used regularly should have some sort of abstraction mechanism for managing repetitive structural operations. For example, a macro of a base boundary could represent base multiplication: $\{A\}$=([$A$][ooooo ooooo]). More complicated constructions will be necessary, although how to create powerful abstraction mechanisms with spatial forms is not clear.

*Tools.* Many people learn and use mathematics with the assistance of computational aids, such as calculators and computers. These tools allow powerful interaction directly with the mathematical objects of standard notation. The boundary calculus suggests that this assistance may hide some of the fundamental dynamics of numbers from those students. Similar computational tools can be built for boundary calculus, wrapping in the preceding techniques for doing mathematics. These tools would need to be linked with powerful functionality that is independent of notation, such as graph plotting.

The boundary calculus does not immediately make mathematics easier to use. Just as numbers are a part of the whole of mathematical understanding, notation is a part of the whole of mathematical use. The supporting cast of tools and techniques must be filled in before the boundary calculus can support full use of mathematics.

## *7.4 Extensions*

Algebra is just the beginning of numerical mathematics: out of algebra stems differential calculus and integral calculus, out of differential calculus stems differential equations and partial differentials, and out of integral calculus stems transformational

Table 7.1: Rules for Boundary Differentiation.

| | |
|---|---|
| D1. | $/(A)/=(A[/A/])$ |
| D2. | $/[A]/=(<[A]>[/A/])$ |
| D3. | $/<A>/=</A/>$ |
| D4. | $/A\ \ B/=/A//B/$ |
| D5. | $//\ \ =$ |

methods including Fourier transforms. The fields continue and boundary calculus can potentially extend to all of them. Boundary numbers may contribute to research and understanding in these areas, just as mathematical notations have enlightened problems in the past [26, 2].

Differentials are rather easily expressed in boundary form. Let $/A/$ represent the differential of $A$. Let $(A)$ be interpreted as $e^A$ and let $[A]$ be interpreted as $\ln a$. The rules defining the inward propagation of the differential operator, shown in Table 7.1, are quite simple. These can be expressed more concisely, as the first two rules are equivalent and the rest are transparencies.

The chain rule, $d(ab) = b\,da + a\,db$, is easily proved from these rules:

| | |
|---|---|
| $/([A][B])/$ | Given |
| $([A][B][/[A]\ \ \ [B]/])$ | D1 |
| $([A][B][/[A]//[B]/])$ | D4 |
| $([A][B][(<[A]>[/A/])(<[B]>[/B/])])$ | D2 (2x) |
| $([A][B][(<[A]>[/A/])])([A][B][(<[B]>[/B/])])$ | Distribution |
| $([A][B]\ \ <[A]>[/A/]\ \ )([A][B]\ \ <[B]>[/B/]\ \ )$ | Involution |
| $([B][/A/])([A][/B/])$ | Inversion |

Boundary numbers lead into deeper mathematics because they have a natural interpretation as exponentials. Many extensions of numbers rely on exponentials, so it is likely that boundary numbers may illuminate some trends that are not directly noticeable in traditional notation. For example, boundary numbers provide a framework for considering the significance of transcendental numbers and their relationships to other numbers.

These extensions would involve translating current mathematical knowledge to boundary form, in search of hidden insight. Boundary numbers provide new perspective on old problems.

### *7.5 Conclusions*

This minimalist effort was motivated by a need to clarify the mathematics we already know. Behind this goal lies a greater purpose of creating a solid foundation for mathematics itself—boundary mathematics has this foundation. Distinction is a fundamental representational act. Networks of nested and collected distinctions can reduce by only a few basic possibilities, because the choices are limited. Different choices direct the initial forms into set theory, natural numbers, or propositional logic. This representational genesis organizes these domains, mapping them from the representational starting point of the void. This clear, explicit organization of mathematical domains is unique to boundary mathematics. It is the study of the most fundamental dynamics of form.

# Chapter 8

# APPLICATIONS

## *8.1 Introduction*

The boundary calculus redefines numbers within the paradigm of boundary mathematics. The calculus may eventually influence the ways we implement and use mathematics, as well as how we teach it.

When solving problems mathematically, we use three basic representational stages (introduced in Chapter 1): the problem in its own terms, the representation of the problem mathematically, and separate representations for performing calculations. These stages combine into a basic solution loop for mathematical activity, shown in Figure 8.1

Traditional notation was designed for the cycle between the top two stages, when representing vaguely understood ideas was more important than the effectiveness of their representation. As mathematics developed, the features of the notation evolved to encompass more and more expressive power. With this progression, calculation with the notation became more complex, to the point where separate forms were introduced to assist in calculation [30]. That is, the notation was no longer suitable for direct calculation, mandating the creation of surrogate forms.

We are taught to compute most arithmetic functions with alternative forms, such as vertical addition and long division. In symbolic manipulation software, algebraic expressions are stored in data structures much different from standard notation, data structures that have been optimized for the type of computation that needs to be done. Computational representations are everywhere; they blend in because they are considered necessary. In fact, they are a discontinuity in representation.

The calculus may simplify the entire solution loop. The critical path goes through the calculation loop, which presently is quite convoluted. The fundamental objects of the calculus are boundaries, computation objects that provide simple and direct computation. The conceptual objects of numbers and arithmetic functions are structures built out of these primitives, rather than being the primitives themselves.

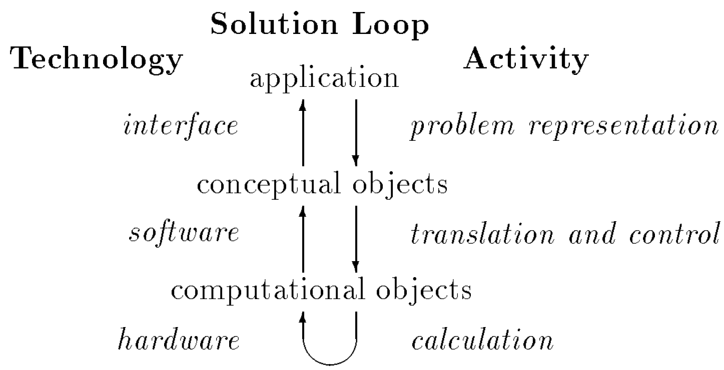



Figure 8.1: Technology and Representation.

With this new set of primitives, the activities in the solution loop become more clearly delineated. In particular, control of the calculation steps, *direction*, becomes clearly separated from the constraints of the calculation, *terrain*. These constraints can be maintained throughout an algorithm so that all intermediate states are mathematically equivalent. These primitive dynamics allow more precise control within the representation than standard forms allow. Any given standard form is not optimally structured for certain calculations.

Pushing the notation down to computational objects significantly affects the tools and mechanisms by which we do mathematics. We can streamline the entire solution loop and the technology that assists us: the hardware that performs calculations, the software that builds and manipulates conceptual objects, our interface to these mathematical objects, and the pedagogy of this entire loop. Each of these areas are profitable applications of this material and are briefly considered in the following sections.

## *8.2 Hardware*

The way we conceptualize mathematics affects how we implement computational hardware. Most machine computations are built upon a system of mathematics based in the higher conceptual objects of the standard notation. The boundary calculus provides lower-level forms from which to base machine computations.

The boundary calculus has computational advantages because it is inherently parallel. Linear computation models have isolated control points which sequence computation—forcing these to be parallel violates their very design, even though mathematics is inherently parallel. The calculus does not limit control and computation but allows as little or as much parallelism as a platform will allow.

To implement the calculus in hardware, a network data structure must be provided in silicon to support the nesting and collecting of boundaries. It must have means of determining equivalent nodes and creating multiple reference. It must support the match and substitute mechanism and it must include a control routine which progresses elements towards a stable, canonical representation.

A number of issues present challenges to this development. How to achieve canonical representations must be determined for the broadest domain of numbers, and constraints must be imposed to remain within that domain. Algorithmic means of avoiding the paradoxes must be fully worked out.

A chip that implements the boundary calculus could be based in the five operations of its definition: set to the void, collect values, wrap with instance, wrap with abstract, or wrap with inverse. The chip could store and manipulate algebraic expressions by manipulating unknowns. Two additional commands provide this function manipulation: create a lambda variable in a register and replace occurrences of a lambda variable with another value.

Ironically, these functions are described for a linear machine. Instruction sets are based on procedural languages within procedural machines. In contrast, this calculus provides a basis for a spatial machine. Thorough architectures of spatial machines have recently been proposed for which the calculus seems compatible [15].

### *8.3 Computer Algebra*

Current computer algebra systems (CAS) adopt data structures which deviate significantly from the parsing structure of standard algebra. These structures help optimize manipulations and operations on functions. The boundary calculus takes this a step further by proposing objects to which specialized network structures can be designed.

To computer algebra, the calculus represents a significant reducing technology. Suddenly, less does more. Fewer formulas are required which ultimately do more. Constraints are easier to follow and simpler to implement. Additionally, the exponential interpretation makes it particularly useful for more customized operations, such as integrals and Fourier transforms where conversion to an exponential form is necessary anyway.

Because the calculus is minimal, it provides a strong basis for automatic theorem proving by limiting decision points. As with hardware application, a key problem is to find a consistent reduction for every expression, using heuristics to guide the application of rules. The decision procedure is easier within the calculus simply because there are fewer constructs to deal with.

The following reductions show that $e^{-\pi/2} = i^i$. In boundary form, the two seemingly different forms actually reduce to an equivalent form.

| | |
|---|---|
| (([<(J [J]([J]<[oo]>))>][(<[oo]>)])) | $e^{-\pi/2}$ |
| (([ (JJ[J]([J]<[oo]>)) ][(<[oo]>)])) | Inverse promotion |
| (( JJ[J]([J]<[oo]>) <[oo]> )) | Involution |
| (( [J]([J]<[oo]>) <[oo]> )) | J cancellation |

| | |
|---|---|
| (([[(([J]<[oo]>))]][(([J]<[oo]>))])) | $i^i$ |
| (( [J]<[oo]> ([J]<[oo]>) )) | Involution |

It would be difficult to implement the calculus in software because it does not cover all the mathematics necessary for a complete mathematics package. The spatial forms are powerful and immediately useful in isolation but they conflict with traditional elements of mathematics. Until more math is similarly expressed in spatial forms, the full power of the paradigm cannot be exploited.

## *8.4 Mathematical Interface*

The calculus makes an immediate contribution to the visualization of mathematics because its spatial constructs afford many visual interpretations. Two interpretations of $ax^2 \rightarrow$ ([a]([[x]][oo])) are shown in Figure 8.2. Because the notation computes in-place, these forms directly support animation.

The boundary calculus defines constructs spatially from the beginning, rather than retrofitting prior structures into spatial form. This frees it from the linear assumptions that corrupt many visual languages.

In breaking from traditional linear methods, the calculus has attained many features that are essential to a good visual language. First it is completely visual. There are no hidden semantics except for the axioms which can be seen in operation and are

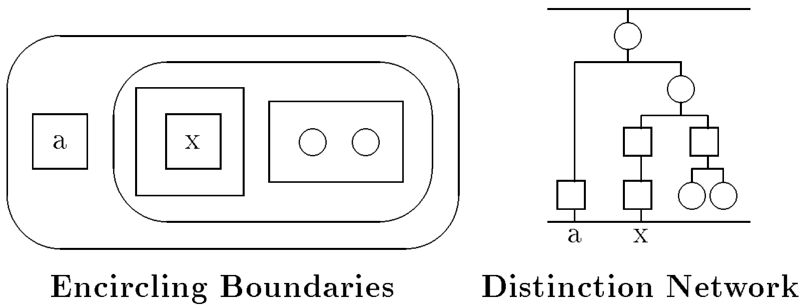



Figure 8.2: Visual Interpretations of Boundary Numbers.

visually simple. No part of the calculus involves dissociated textual code; the picture itself completely and unambiguously describes a mathematical expression.

The calculus also performs computation in-place, a valuable feature of a visual programming language [18]. Expressions change themselves into new forms, exactly where they are, in many cases simply by fading in or out parts of the expression. This computation is naturally parallel and concurrent, without explicit control points: computation occurs where and when it can.

The boundary calculus can improve the mathematical interface by making it visual: expressions are visually specified and computation occurs visually on the same forms.

### *8.5 Education*

The standard algebraic notation—made of operators and equations—dominates mathematical experience. As the primary means of communicating algebraic concepts, it is used in textbooks, on flash cards and posters, on tests and homework. The notation holds an exclusive role: it is the only common thread through all mathematical experience and so is commonly confused as being mathematics itself.

The notation not only serves to implement mathematical concepts, it also serves as an authority on how these concepts are made to work. The notation links ideas into common abstractions that serve as the identifying references for the ideas. All mathematical explanations are ultimately tied back to these representations; they are relied upon pedagogically as fundamental to mathematics.

Standard notation conceals knowledge of the relationships between functions. Formulas specify how they transform: some functions have several representations while others are unique and some combinations can be simplified while others cannot. Because the visual form does not convey these relationships, much external knowledge must be brought to the task of doing mathematics.

In this way, standard notation fails to support learning mathematics. Availability and application of rules are not suggested by the forms. Knowledge of these rules remains distinct and separate from this mathematical interface. Dissociating knowledge from appearance leaves an impression that the knowledge is arbitrary and without foundation. Only by accepting this knowledge on faith can a student bypass the contradictions of form introduced by this dissociation.

The failure to instantiate mathematical knowledge in its visual forms further contributes to the misconception that mathematics is a purely cognitive activity. In doing so, mathematical ability is politely restricted to only the "gifted" population, another failure of mathematics education.

The boundary calculus can impact mathematics education. It provides another reference for the implementation of formal mathematical concepts and arguably presents a better interface to mathematical behavior.

### *8.6 Conclusions*

The stages of mathematical representation can be readdressed under the paradigm of boundary mathematics so that the activities of mathematics are more clearly delineated. The boundary calculus isolates manipulative constraints, leaving conceptual objects as constructions of the boundary forms. The delineation gives those particular constructions meaning and benefit that is usually hidden. Standard choices for organizing numbers are actually optimized for certain criteria. Opening up representation with the boundary calculus makes the criteria of representation and the choices of structure more explicit.

# GLOSSARY

**abstract** A type of boundary in the calculus of number. Abstract forms the *black hole* when empty. Written as $\square$ when empty or `[`$A$`]` around boundary configuration $A$.

**anti-logarithm** Functional interpretation of the instance boundary, $e^x \rightarrow$ `(`$x$`)`.

**black hole** The abstract boundary with no contents. The black hole dominates its context. Written as $\square$.

**boundary** Drawing of a distinction. `()`, `[]`, and `<>` are types of boundaries.

**boundary mathematics** A representational and computational paradigm based on boundary forms that transform by match and substitute.

**cardinality** A theorem stating that a repeated form can be rewritten using a single reference. Written as $A \ldots A$=`([`$A$`][`$\circ \ldots \circ$`])`.

**collection** Spatial justaposition of boundary configurations, as $AB$. Collections are unordered, ungrouped, and variary.

**configuration** Any algebraic boundary expression, including void.

**content** The space inside of a distinction. e.g. `(`*content*`)`.

**context** The space outside of a distinction. e.g. *context*`()`.

**distinction** A cleaving of space. Drawn as a boundary, `()`.

**distribution** An axiom that defines the modifier form as distributive over collection. Written as `(`$A$`[`$BC$`])`$\doteq$`(`$A$`[`$B$`])(`$A$`[`$C$`])`.

**dominion** A theorem stating that elements collected with the black hole are irrelevant. Written as $\Box a = \Box$.

**e** A transcendental intepretation of nested instance boundaries, $e \rightarrow (\circ)$.

**equivalence axiom** Two or more templates that define equivalent expressions. e.g. $([A]) \doteq A$.

**fractional cardinality** Expression of a partial cardinality (inverse of repeated cardinality). Written as $([A]<[\circ \ldots \circ]>)$ where the number of instance marks indicates the demonimator of the reciprocal multiplying $A$.

**i** A complex number formed with a half cardinality of J. Written as $i \rightarrow (([J]<[\circ\circ]>))$.

**instance** A type of boundary in the calculus of number. Instance forms the *unit* when empty. Written as $\circ$ when empty or $(A)$ around boundary configuration $A$.

**inverse** A type of boundary in the calculus of number. Inverse serves as a generalized inverse. Written as $\triangle$ when empty or $<A>$ around boundary configuration $A$.

**inverse cancellation** A theorem stating that inverse applied twice cancels out. Written as $<<A>> = A$.

**inverse collection** A theorem stating that a collection of two inverted elements is equivalent to the inversion of the collected elements. Written as $<A><B> = <AB>$.

**inverse promotion** A theorem stating that inverse can be brought inside of the modifier form. Written as $<(A[B])> = (A[<B>])$.

**inversion** An axiom that defines the generalized inverse. Written as $A<A> \doteq$ .

**involution** An axiom that defines instance and abstract as functional inverses. Written as $([A]) \doteq A \doteq [(A)]$.

**J** The phase element, $J \doteq [<\circ>]$.

**J cancellation** A theorem stating that J cancels with itself. Written as $JJ=$ .

**match and substitute** Replacement of template variables in an equivalence axiom so that one of its templates matches part of a given expression.

**modifier form** A configuration composed of instance around abstract. Written as $(A[B])$, with $A$ part of the form, wrapped around contents $B$.

**natural logarithm** Functional interpretation of the abstract boundary, $\ln(a) \rightarrow [a]$.

**negative cardinality** Expression of an inverted repetition. Written for reference $A$ as $([A][<\circ\ldots\circ>])$, where the number of units indicates the magnitude of the cardinality.

**phase element** A form that captures the inverse in an arithmetic construction. Defined as $J \doteq [<\circ>]$. It serves as the inverse when placed in the modifier form, as $(J[A])=<A>$.

**phase independence** A theorem stating that the contents of the phase operator can be moved to the context. Written as $[<(A)>]=A[<()>]$.

**phase operator** A contruction nesting the abstract, inverse, and instance boundaries around an argument. Written as $[<(A)>]$.

**pi** An algebraic construction of the radian unit. Written as $\pi \rightarrow (J[J]([J]<[\circ\circ]>))$.

**template** A boundary expression that includes zero or more template variables, e.g. $(A[B])$.

**template variable** A form found in templates that can be replaced by any boundary expression. Template variables are written as uppercase letters.

**unit** The instance boundary with no contents. It serves as the unit in cardinality. Written as $\circ$.

**void** Empty space; a lack of structure upon which distinctions are made.

# BIBLIOGRAPHY


[1] B. Banaschewski. On G. Spencer Brown's Laws of Form. *Notre Dame Journal of Formal Logic*, 18:507–509, 1977.

[2] C.B. Boyer. *A History of Mathematics*. John Wiley and Sons, Inc., New York, 1991.

[3] W. Bricken and E. Gullichsen. An introduction to boundary logic with the losp deductive engine. *Future Computing Systems*, 2(4), 1989.

[4] W. Bricken and P.C. Nelson. Pure lisp as a network of systems. In *Proceedings of the Second Kansas Conference: Knowledge-Based Software Development*, Kansas State University, 1986.

[5] W. Bricken. Boundary numbers. Technical report, Advanced Decision Systems, 1987.

[6] W. Bricken. The efficiency of boundary mathematics for deduction. Technical Report ADS-6824-1, Advanced Decision Systems, 1987.

[7] W. Bricken. Implementation of the extended program model for the intelligent program editor. Technical Report TR-1047-03, Advanced Decision Systems, 1987.

[8] W. Bricken. Utilizing boundary mathematics for deduction. Technical report, Advanced Decision Systems, 1987.

[9] W. Bricken. Boundary mathematics, 1993. Personal Communication.

[10] G. Spencer Brown. An algebra for the natural numbers. Personal notes, May 1961.

[11] G. Spencer Brown. *Laws of Form*. Bantam, New York, 1969.

[12] J.H. Conway. *On Numbers and Games*. Academic Press, New York, 1976.

[13] P. Cull and W. Frank. Flaws of form. *International Journal of General Systems*, 5(4):201–211, 1979.

[14] B. Doyle, M. Friedman, and B. York. An introduction to forms and logic. Technical Report BUCS-87-008, Boston University, 1987.

[15] Y. Feldman and E. Shapiro. Spatial machines: a more realistic approach to parallel computation. *Communications of the ACM*, 35(10):60–73, 1992.

[16] J.A. Goguen and F.J. Varela. Systems and distinctions; duality and complementary. *International Journal of General Systems*, 5:31–43, 1979.

[17] J. James and W. Bricken. A boundary notation for visual mathematics. In *Proceedings of the IEEE Workshop on Visual Languages*, 1992.

[18] K.M. Kahn and V.A. Saraswat. Complete visualizations of concurrent programs and their executions. In *Proceedings of the IEEE Workshop on Visual Languages*, pages 7–15, 1990.

[19] L.H. Kauffman. Demorgan algebras - completeness and recursion. In *Proceedings of the Eighth International Symposium on Multiple-Valued Logic*, pages 82–86. IEEE Computer Society Press, 1978.

[20] L.H. Kauffman. Network synthesis and Varela's calculus. *International Journal of General Systems*, 4:179–187, 1978.

[21] L.H. Kauffman. Self-reference and recursive forms. *Journal of Social and Biological Structures*, 10:53–72, 1978.

[22] L.H. Kauffman. Formal arithmetic. Personal communication to W. Bricken, August 1986.

[23] L.H. Kauffman. Reinventing the negative numbers. Personal communication to W. Bricken, June 1987.

[24] L.H. Kauffman. The form of arithmetic. In *Proceedings of the 18th International Symposium on Multiple-Valued Logic*. IEEE Computer Society Press, 1988.

[25] L.H. Kauffman and F.J. Varela. Form dynamics. *Journal of Social and Biological Structures*, 3:171–206, 1980.

[26] M. Kline. *Mathematical Thought From Ancient To Modern Times*. Oxford University Press, New York, 1972.

[27] L.J. Kohout and V. Pinkava. The algebraic structure of the Spencer Brown and Varela calculi. *International Journal of General Systems*, 6:155–171, 1980.

[28] L. Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley Publishing Company, Menlo Park, CA, 1986.

[29] National Council of Teachers of Mathematics, Inc., Reston, Va. *Curriculum and Evaluation Standards for School Mathematics*, 1989.

[30] R.S. Nickerson. Counting, computing, and the representation of numbers. *Human Factors*, 30(2):181–199, 1988.

[31] R.A. Orchard. On the laws of form. *International Journal of General Systems*, 2:99–106, 1975.

[32] D.G. Schwartz. Isomorphisms of Spencer Brown's Laws of Form and Varela's calculus for self-reference. *International Journal of General Systems*, 6:239–255, 1981.

[33] R.G. Shoup. A complex logic for computation with simple interpretations for physics, 1992. Personal communication.

[34] F.J. Varela. A calculus for self-reference. *International Journal of General Systems*, 2:5–24, 1975.

[35] F.J. Varela. *Principles of Biological Autonomy.* North Holland, New York, 1979.

[36] F.J. Varela and J.A. Goguen. The arithmetic of closure. *Journal of Cybernetics*, 8:291–324, 1978.

Appendix A

# CONVERSION

## *A.1 Numbers*

The basic numbers types can be expressed in boundary notation. For each number type below, the general structure of that type is given in boundary form followed by some examples.

| | **Standard** | **Boundary** |
|---|---|---|
| *Zero* | $0$ | |
| *Natural* | $n$ | o...o |
| | $1$ | o |
| | $2$ | oo |
| *Integer* | $i$ | $n$ or <$n$> |
| | $-1$ | <o> |
| | $-2$ | <oo> |
| *Rational* | $q$ | ([$i_1$]<[$i_2$]>) |
| | $2/3$ | ([oo]<[ooo]>) |
| | $1/4$ | (<[oooo]>) |
| *Algebraic Irrational* | $r$ | (([[$q_1$]][$q_2$])) or $r_1$([$r_2$]([[$r_3$]][$r_4$])) |
| | $\sqrt[3]{7}$ | (([[ooooooo]]<[ooo]>)) |
| | $\sqrt{2}/2$ | (([[oo]]<[oo]>)<[oo]>) |
| *Complex* | $c$ | $r_1$([$r_2$]([$J$]<[oo]>)) |
| | $i$ | (([$J$]<[oo]>)) |
| | $2+4i$ | oo([oooo]([$J$]<[oo]>)) |
| *Transcendental* | $e$ | (o) |
| | $\pi$ | ($J$[$J$]([$J$]<[oo]>)) |
| | $i\pi$ | [<o>] |

## *A.2 Functions*

The standard algebraic functions of addition, subtraction, multiplication and division can be formed with boundaries. Powers and transcendental functions can be formed with the same objects. These are shown below.

| | **Standard** | **Boundary** |
|---|---:|---|
| *Identity* | $x$ | $x$ |
| *Inverse* | $-x$ | $<x>$ |
| | $1/x$ | $(<[x]>)$ |
| *Addition* | $x+y$ | $xy$ |
| | $x-y$ | $x<y>$ |
| *Multiplication* | $x \times y$ | $([x][y])$ |
| | $x/y$ | $([x]<[y]>)$ |
| *Power* | $x^y$ | $(([[x]][y]))$ |
| | $x^{-y}$ | $(([[x]][<y>]))$ |
| | $\sqrt[y]{x}$ | $(([[x]]<[y]>))$ |
| *Transcendental* | $e^x$ | $(x)$ |
| | $\ln x$ | $[x]$ |

## *A.3 Number Formats*

The boundary notation does not restrict the structural format of numbers. The notation can express numbers in a variety of formats by adapting the algebraic structure of that format. Below are some examples.

Define $b$ to be the base radix, $b \doteq \circ\circ\circ\circ\circ\ \circ\circ\circ\circ\circ$.

Define $\{A\}$ to be the base boundary, $\{A\} \doteq ([b][A])$.

| | Standard | Boundary |
|---|---|---|
| *Mixed* | $i_0\frac{i_1}{i_2}$ | $i_0$([$i_1$]<[$i_2$]>) |
| | $4\frac{2}{3}$ | oooo([oo]<[ooo]>) |
| | $1\frac{1}{2}$ | o(<[oo]>) |
| *Based* | $\ldots + i_1b^1 + i_0b^0$ | {{...}$i_1$}$i_0$ |
| | 17 | {o}ooooo oo |
| | 6243 | {{{oooooo}oo}oooo}ooo |
| *Scientific* | $r \times 10^i$ | ([$r$]([[$b$]][$i$])) |
| | $3 \times 10^8$ | ([ooo]([[$b$]][oooooooo])) |
| | $6 \times 10^{23}$ | ([oooooo]([[$b$]][{oo}ooo])) |

## *A.4 Standard Postulates*

The standard field postulates can be derived directly from the boundary axioms. Derivations of these postulates are shown below.

*Commutativity*

$x + y = y + x$    $xy$=$yx$

$xy = yx$    ([$x$][$y$])=([$y$][$x$])

Commutativity is implicit in the boundary forms because no ordering has been imposed in the first place. The typographic difference between $xy$ and $yx$ is not recognized by the boundary notation: containment is meaningful, ordering is not.

*Associativity*

$x + (y + z) = (x + y) + x$    $xyz$=$xyz$

$x(yz) = (xy)z$    ([$x$][([$y$][$z$])])=([([$x$][$y$])][$z$])

Associativity is similarly implicit in the boundary forms. Addition by collection is a variary operation, allowing zero or more items to be collected at a time. Collecting with a collection leaves no evidence of functional ordering. Multiplication does leave evidence of functional ordering but this can be erased and replaced by involution.

*Distribution*

$x(y + z) = (xy) + (xz)$    ([$x$][$yz$])=([$x$][$y$])([$x$][$z$])

Distribution of multiplication over addition follows as a special case of the distribution axiom.

*Identity*

$$\begin{array}{rl} x + 0 = x & x{=}x \\ 1x = x & ([\circ][x]){=}([x]){=}x \end{array}$$

The additive identity is implicit in spatial collection as the void. The multiplicative identity reduces to void within the outer instance boundary and the resulting unary multiplication reduces to identity.

## A.5 Formulas

Common algebraic formulas can be deduced directly from the boundary axioms. Some are listed below. Many addition formulas have corresponding ones in multiplication. The pairs are listed together to demonstrate their similarities in boundary form.

*Cardinality*

$$\begin{array}{rl} x + \ldots + x = nx & x\ldots x{=}([x][\circ\ldots\circ]) \\ x \times \ldots \times x = x^n & ([x]\ldots[x]){=}(([[x]][\circ\ldots\circ])) \end{array}$$

Cardinality counts repeated instances in the same space, whether a collection of $x$ for addition or a collection of $[x]$ for multiplication.

*Inversion*

$$\begin{array}{rl} x + (-x) = 0 & x<x>{=} \\ x/x = 1 & ([x]<[x]>){=}() \end{array}$$

Inversion cancels an item and its inverse. It acts on the additive inverse, cancelling $x$ and $<x>$, and it acts on the multiplicative, cancelling $[x]$ and $<[x]>$.

*Inverse Collection*

$$\begin{array}{rl} (-x) + (-y) = -(x + y) & <x><y>{=}<xy> \\ 1/x \times 1/y = 1/xy & ([(<[x]>)][(<[y]>)]){=}(<[x][y]>) \end{array}$$

Inverse collection accumulates multiple objects within the same inverse boundary. It carries the additive inverse over an sum and it carries the multiplicative inverse over a product.

*Inverse Cancellation*

$$\begin{aligned} -(-x) &= x & <<x>> &= x \\ 1/(1/x) &= x & (<[(<[x]>)]>) &= x \end{aligned}$$

The generalized inverse cancels itself out.

*Inverse Promotion*

$$-x = (-1)x \qquad <x> = ([x][<\circ>])$$

$$\begin{aligned} 1/x &= x^{-1} & (<[x]>) &= (([[x]][<\circ>])) \\ -(xy) &= x(-y) & <([x][y])> &= ([x][<y>]) \\ 1/x^y &= x^{-y} & (<[(([[x]][y]))]>) &= (([[x]][<y>])) \end{aligned}$$

Inverse promotion converts the additive inverse to a product with negative one and the multiplicative inverse to a power of negative one. It provides a straightforward means of moving the inverse around within an expression.

*Powers*

$$\begin{aligned} x^1 &= x & (([[x]][\circ])) &= x \\ x^0 &= 1 & (([[x]][])) &= () \\ x^m x^n &= x^{m+n} & ([(([[x]][m]))][(([[x]][n]))]) & \\ & & = (([[x]][mn])) & \\ (x^m)^n &= x^{mn} & (([[(([[x]][m]))]][n])) & \\ & & = (([[x]][([m][n])])) & \\ x^m y^m &= (x \times y)^m & ([(([[x]][m]))][(([[y]][m]))]) & \\ & & = (([[([x][y])]][m])) & \end{aligned}$$

Powers of one and zero reduce in boundary form by involution and dominion. The other power formulas are derived by involution and distribution.

*Logarithms*

$$\begin{aligned} \ln(xy) &= \ln x + \ln y & [([x][y])] &= [x][y] \\ \ln x/y &= \ln x - \ln y & [([x]<[y]>)] &= [x]<[y]> \\ \ln x^y &= y \ln x & [(([[x]][y]))] &= ([[x]][y]) \end{aligned}$$

These logarithmic formulas reduce directly by involution.

*Radians*

$1 + e^{i\pi} = 0$ ○([<○>]) =

Euler's formula works because the form [<○>] was defined that way.

## Appendix B

# EXAMPLES

### *B.1 Multiplication*

Here is an example of multiplication of two integers. To visually simplify the forms, a few definitions will be made first.

Let $b$ be the base radix, $b \doteq$ ooooo ooooo. Let $\{A\}$ be the base function on $A$, defined as $\{A\} \doteq$ ([$b$][$A$]). The radix carries into the base function, $b\{A\} = \{\circ A\}$. The base function collects, $\{A\}\{B\} = \{AB\}$, and promotes, $\{$($A$[$B$])$\}$=($A$[$\{B\}$]), just like inverse:

$$\{A\}\{B\}=([b][A])([b][B])=([b][AB])=\{AB\},$$

$$\{(A[B])\}=([b][(A[B])])=([b]A[B])=(A[([b][B])])=(A[\{B\}]).$$

Using these definitions, the multiplication $23 \times 114$ can be computed by making copies at each magnitude, collecting magnitudes, and doing a carry operation.

| | |
|---|---|
| 23 × 114 | Given |
| {oo}ooo × {{o}o}oooo | Number Rewrite |
| ([{oo}ooo][{{o}o}oooo]) | Function Rewrite |
| ([{oo}][{{o}o}oooo])([ooo][{{o}o}oooo]) | Distribution |
| {([oo][{{o}o}oooo])}([ooo][{{o}o}oooo]) | Promotion |
| {{{o}o}oooo{{o}o}oooo}{{o}o}oooo{{o}o}oooo{{o}o}oooo | Cardinality (2x) |
| {{{o}o {o}o}oooo oooo {o}o {o}o {o}o}oooo oooo oooo | Collection |
| {{{o o} o o  o o o} oooo oooo o o o}oooo oooo oooo | Collection |
| {{{oo}ooooo}*b*o}*b*oo | Replacement |
| {{{oo}oooooo}oo}oo | Carry |
| 2622 | Rewrite |

### *B.2 Square Root*

The following square root calculation is an *informed* calculation. It does not represent an algorithm for determining square roots, it merely demonstrates the equivalence of $\sqrt{9}$ and 3.

| | |
|---|---|
| $\sqrt{9}$ | Given |
| (([[ooooooooo]]<[oo]>)) | Rewrite |
| (([[([ooo][ooo])]]<[oo]>)) | Cardinality, $A$=ooo |
| (([[(([[ooo]][oo]))]]<[oo]>)) | Cardinality, $A$=[ooo] |
| (( [[ooo]][oo] <[oo]>)) | Involution (2x) |
| (( [[ooo]] )) | Inversion |
| ooo | Involution (2x) |
| 3 | Rewrite |

### *B.3 Fractions*

In standard notation, fractions are added by forming common denominators and distributing. In the boundary calculus, they are combined by forming common modifiers and distributing.

| | |
|---|---|
| $1/a + 1/b$ | Given |
| (<[*a*]>)(<[*b*]>) | Rewrite |
| ([*b*]<[*b*]><[*a*]>)([*a*]<[*a*]><[*b*]>) | Inversion |
| ([*ab*]<[*a*]><[*b*]>) | Distribution of <[*a*]><[*b*]> |
| ([*ab*]<[*a*][*b*]>) | Inverse Collection |
| ([*ab*]<[([*a*][*b*])]>) | Involution |
| $(a + b)/(ab)$ | Rewrite |

### *B.4 Algebraic Distribution*

This algebraic derivation uses the distribution axiom in ways that appear different when written in standard form, distributing $a+b$ over subtraction $a-b$ and distributing $a$ and $-b$ over addition $a+b$. These applications all use the same boundary rule. Expansion of this product can be done in different ways, this ordering and choice of distributions are but one way.

| | |
|---|---|
| $(a+b)(a-b)$ | Given |
| ([ab][a<b>]) | Rewrite |
| ([ab][a])([ab][<b>]) | Distribution of [ab] |
| ([a][a])([b][a])([ab][<b>]) | Distribution of [a] |
| ([a][a])([b][a])([a][<b>])([b][<b>]) | Distribution of [<b>] |
| ([a][a])([b][a])<([a][b])><([b][b])> | Inverse Promotion (2x) |
| ([a][a])                 <([b][b])> | Inversion |
| (([[a]][○○])) <([[b]][○○]))> | Cardinality (2x) |
| $a^2 - b^2$ | Rewrite |

In the boundary calculus, operations can be done in parallel. Here simultaneous operations of the same kind are done in parallel and marked with "(2x)". The second and third distribution operations can be done in parallel as can the final inversion and cardinality.

### *B.5 Quadratic Formula*

The quadratic formula gives solutions to equations of the form $ax^2+bx+c=0$. To derive the quadratic formula in the boundary calculus, this equation will be transformed to this form, which reveals the solutions of $x$:

$$(x - \frac{-b+\sqrt{b^2-4ac}}{2a})(x - \frac{-b-\sqrt{b^2-4ac}}{2a}) = 0$$

The above quadratic appears in boundary form as the following void equivalent:

([a]([[x]][○○]))([b][x])c =

The objective is to transform it to the following form, revealing solutions, $s_1$ and $s_2$:

([x<$s_1$>][x<$s_2$>]) =    .

To remove the first coefficient $a$, introduce its multiplicative inverse from the void.

| | |
|---|---|
| ([]) | Involution |
| ([]<[a]>) | Dominion |

Then substitute the quadratic in the black hole of the product.

| | |
|---|---|
| ([ ([a]([[x]][○○]))([b][x])c ]<[a]>) | Void substitution |

The $a$ coefficient can now be removed. Distribute its inverse and cancel.

| | |
|---|---|
| ([([a]([[x]][∘∘]))]<[a]>)([([b][x])]<[a]>)([c]<[a]>) | Distribution |
| ( [a]([[x]][∘∘]) <[a]>)( [b][x] <[a]>)([c]<[a]>) | Involution |
| ( ([[x]][∘∘]) )( [b][x] <[a]>)([c]<[a]>) | Inversion |

Now the $x^2$ has a unary coefficient. Define $d$ and $e$.

d ≐ ([b]<[a][∘∘]>)

e ≐ ([c]<[a]>)

Work them into the equation.

| | |
|---|---|
| (([[x]][∘∘]))([x][b]<[a]><[∘∘]>[∘∘]) ([c]<[a]>) | Inversion |
| (([[x]][∘∘]))([x][b]<[a] [∘∘]>[∘∘]) ([c]<[a]>) | Collection |
| (([[x]][∘∘]))([x][([b]<[a][∘∘]>)][∘∘])([c]<[a]>) | Involution |
| (([[x]][∘∘]))([x][d][∘∘])e | Definition of $d$ and $e$ |

Flatten and complete the square.

| | |
|---|---|
| ([x][x])([x][d][∘∘])e | Cardinality |
| ([x][x])([x][d][∘∘])([d][d])<([d][d])>e | Inversion |

Now factor it.

| | |
|---|---|
| ([x][x])([x][d])([x][d])([d][d])<([d][d])>e | Cardinality |
| ([x][xd])([xd][d])<([d][d])>e | Distribution (2x) |
| ([xd][xd])<([d][d])>e | Distribution (2x) |

Make a difference of squares.

| | |
|---|---|
| ([xd][xd])<([d][d])><<e>> | Inverse Cancellation |
| ([xd][xd])<([d][d])<e>> | Collection |
| ([xd][xd])<(([[([d][d])<e>]]))> | Involution |
| ([xd][xd])<(([[([d][d])<e>]]<[∘∘]>[∘∘]))> | Inversion |
| ([xd][xd])<(([[(([[([d][d])<e>]]<[∘∘]>))]][∘∘]))> | Involution |

Define $f$ to clean up the negative square.

f ≐ (([[([d][d])<e>]]<[∘∘]>))

Put it into the equation and flatten to a product.

| | |
|---|---|
| ([xd][xd])<(([[f]][∘∘]))> | Definition of $f$ |
| ([xd][xd])<([f][f])> | Cardinality |

Separate into two factors.

$([xd][xd])([xd][f])<([xd][f])><([f][f])>$ Inversion
$([xd][xd])([xd][f])([xd][<f>])([f][<f>])$ Promotion
$([xd][xdf])([xdf][<f>])$ Distribution
$([xdf][xd<f>])$ Distribution
$([x<<df>>][x<<d<f>>>])$ Inverse Cancellation

This gives two solutions.

$x = <df>$ and $x = <d<f>>$

Expand these for the recognized solutions. Recall $f$.

$f \doteq (([[([d][d])<e>]]<[\circ\circ]>))$

First expand the interior of $f$.

$([d][d])<e> =$
$= ([([b]<[a][\circ\circ]>)][([b]<[a][\circ\circ]>)])<e>$ Definition of $d$
$= (\quad [b]<[a][\circ\circ]> \quad [b]<[a][\circ\circ]> \quad )<e>$ Involution
$= ([b]<[a][\circ\circ]>[b]<[a][\circ\circ]>)<([c]<[a]>)>$ Definition of $e$
$= ([b]<[a][\circ\circ]>[b]<[a][\circ\circ]>)$
$\quad <([c]<[a]><[a][\circ\circ][\circ\circ]>[a][\circ\circ][\circ\circ])>$ Inversion
$= ([b][b]<[a][\circ\circ][a][\circ\circ]>)$
$\quad <([c]<[a][a][\circ\circ][\circ\circ]>[a][\circ\circ][\circ\circ])>$ Collection
$= ([([b][b])]<[a][\circ\circ][a][\circ\circ]>)$
$\quad <(<[a][a][\circ\circ][\circ\circ]>[([c][a][\circ\circ][\circ\circ])])>$ Involution
$= ([([b][b])]<[a][\circ\circ][a][\circ\circ]>)$
$\quad <(<[a][a][\circ\circ][\circ\circ]>[<([c][a][\circ\circ][\circ\circ])>])$ Promotion
$= (<[a][\circ\circ][a][\circ\circ]>[([b][b])<([c][a][\circ\circ][\circ\circ])>])$ Distribution
$= (<([[a][\circ\circ]][\circ\circ])>[(([[b]][\circ\circ]))<([c][a][\circ\circ][\circ\circ])>])$ Cardinality
$= (<([[a][\circ\circ]][\circ\circ])>[(([[b]][\circ\circ]))<([c][a][\circ\circ\circ\circ])>])$ Cardinality
$= (([<[a][\circ\circ]>][\circ\circ])[(([[b]][\circ\circ]))<([c][a][\circ\circ\circ\circ])>])$ Promotion

Then determine $f$.

| | |
|---|---|
| f = ((<[oo]>[[(([<[a][oo]>][oo]) | |
| [(([[b]][oo]))<([c][a][oooo])>])]])) | Substitution |
| = ((<[oo]>[ ([<[a][oo]>][oo]) | |
| [(([[b]][oo]))<([c][a][oooo])>] ])) | Involution |
| = ((<[oo]>[ ([<[a][oo]>][oo]) ]) | |
| (<[oo]>[[(([[b]][oo]))<([c][a][oooo])>]])) | Distribution |
| = ((<[oo]> [<[a][oo]>][oo] ) | |
| (<[oo]>[[(([[b]][oo]))<([c][a][oooo])>]])) | Involution |
| = (( [<[a][oo]>] ) | |
| (<[oo]>[[(([[b]][oo]))<([c][a][oooo])>]])) | Inversion |
| = (<[a][oo]>(<[oo]>[[(([[b]][oo]))<([c][a][oooo])>]])) | Involution |

Define g as the root expression.

g ≐ ((<[oo]>[[(([[b]][oo]))<([c][a][oooo])>]]))

The solutions are a function of g.

| | |
|---|---|
| <df> = <([b]<[a][oo]>)(<[a][oo]>[g])> | Definition of g |
| = <([bg]<[a][oo]>)> | Distribution |
| = ([<bg>]<[a][oo]>) | Promotion |
| = ([<b><g>]<[a][oo]>) | Collection |

| | |
|---|---|
| <d<f>> = <d><<f>> | Collection |
| = <d>f | Inverse Cancellation |
| = <([b]<[a][oo]>)>(<[a][oo]>[g]) | Definition of g |
| = ([<b>]<[a][oo]>)(<[a][oo]>[<g>]) | Promotion |
| = ([<b>g]<[a][oo]>) | Distribution |

The solutions to the quadratic formula, ([a](([[x]][oo]))([b][x])c, are:

x = ([<b>((<[oo]>[[(([[b]][oo]))<([oooo][c][a])>]]))]<[oo][a]>),

x = ([<b><((<[oo]>[[(([[b]][oo]))<([oooo][c][a])>]]))>]<[oo][a]>).

which translate to

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad and \quad x = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$