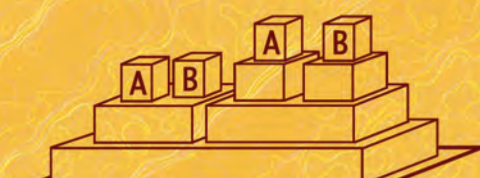
Unary Logic

Volume I
BOUNDED and ILLUSORY FORMS

William Bricken

a postsymbolic logic for the twenty-first century



Unary Logic

Volume I

Any comments, corrections, refinements or suggestions you may have will be greatly appreciated.

You can reach me at
wm@iconicmath.com

Thanks.

AI free: Unary logic and the content of this volume were created solely by biological intelligence.

In this series, available from Amazon Books:

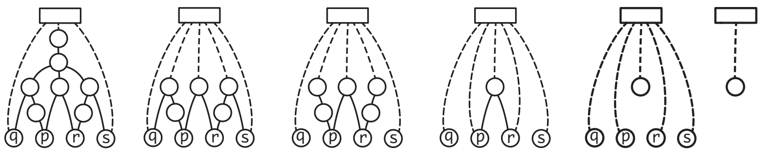
Unary Logic (2026)
Volume I   Bounded and Illusory Forms
ISBN: 978-1-7324851-6-7
9-781732-485167

Unary Logic (projected for 2028)
Volume II   Software and Hardware Applications
ISBN: not yet allocated

Unary Logic (unlikely)
Volume III   All the Other Stuff

Unary Logic Volume I

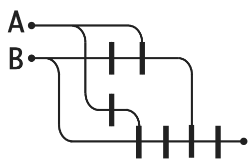
ISBN (soft-cover edition): 9-781732-485167
Unary Press, an imprint of
Unary Computers, Snohomish Washington USA



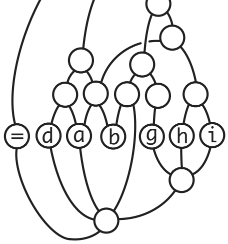
Unary Logic

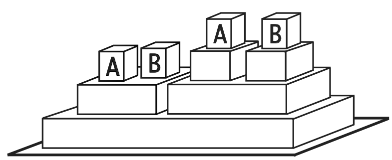
Volume I
BOUNDED and ILLUSORY FORMS

William Bricken



a postsymbolic logic for the twenty-first century





Chapters Volume I

Contents

List of Figures

Cast of Characters

Visionaries

Aristotle
George Boole
Gregory Bateson
Lewis Carroll
Heinz von Foerster
Gottlob Frege
Louis Kauffman
Gottfried Leibniz
Charles Sanders Peirce
Bertrand Russell
George Spencer-Brown
Francisco Varela
Alfred North Whitehead
Ludwig Wittgenstein
Stephen Wolfram

Special Thanks to

Paul Allen
Cliff Barney
Colin & Ian Bricken
Andrew Crompton
Graham Ellsbury
Heinz von Foerster
Sally Heilstedt
Jeffrey James
Louis Kauffman
Jaron Lanier
John McCarthy
Meredith Bricken Mills
Amy Morrison
Ingram Olkin
Frank Ruskey
Daniel Shapiro
Richard Shoup
Spencer-Brown Society
John Walker
Richard Weyhrauch
Stephen Wolfram

Research and Perspective

Jon Barwise & John Etchemendy
Gregory Bateson
Frank M. Brown
George Burnett-Stuart
Arthur Collings
Frithjof Dau
Pierre Duhem
Jack Enstrom
Thomas Etter
Henning Genz
Joseph Goguen
Mark Greaves
Douglas Hofstadter
Sabine Hossenfelder
Jeffrey James
Virginia Klenk
W. & M. Kneale
Georg Kreisel
George Lakoff & Rafael Núñez
Jaron Lanier
E. J. Lemmon
Jan Łukasiewicz
Hōsaku Matsuo
Tor Nørretranders
Francis Pelletier
Ahti-Veikko Pietarinen
W. V. Quine
Don Roberts
Brian Rotman
Frank Ruskey
Jan Simner
Sun-Joo Shin
Richard Shoup
Herbert Simon
John Sowa
Patrick Suppes
Paul Watzlawick
J, Jay Zeman

Colleagues

John Awbrey
Cliff Barney
George Burnett-Stuart
Arthur Collings
Leon Conrad
Andrew Crompton
Randolph Dible
Graham Ellsbury
Jack Engstrom
Thomas Etter
Fred Furtek
Eric Gullichsen
Nat Hellerstein
Robert Horn
Jeffrey James
Louis Kauffman
David Keenan
Valerie Kim
Moshe Klein
Sam Knight
Kerry Koitzsch
Peter Kramer
Bernie Lewin
Thomas McFarlane
Philip Meguire
Armahedi Mehzar
Meredith Bricken Mills
Daniel Shapiro
Richard Shoup
Andrew Singer
Ivo Velitchkov
Kurt Von Meyer
William Winn

Historical Figures	***circa***	***chapter***
Parmenides	500 BCE	3, 9
Democritus	400	9
Euclid	400	2, 3, 14
Plato	360	3
Aristotle	320	1-3, 6, 8, 9, 14, 15
Chrysippus	200	15
Zhuang Zhou	200 BCE	9
Augustine	400 CE	9, 11
Ramon Llull	1300	2
William of Ockham	1300	1
Chaucer	1400	9
DaVinci	1500	9
G. Reisch	1500	14
A. Veracruz	1560	14
Galileo	1620	3, 9
Leibniz	1700	2, 3, 5, 10
Euler	1760	2, 14
Kant	1800	11
Lagrange	1800	2
Boole	1850	2, 3, 6-8, 10
Lewis Carroll	1880	11, 12
Frege	1900	1, 2, 3, 6, 8, 10, 14
Hilbert	1900	2, 3, 6, 8, 15
Peirce	1900	1-4, 6-10, 14, 15
Einstein	1940	1, 2, 3
Gödel	1940	6
Matisse	1940	3
Russell	1940	2, 3, 6, 8, 11,15
Turing	1940	3, 6
Whitehead	1940	2, 6, 8, 11, 15
Wittgenstein	1940	2, 6, 7, 9, 10

Preface

...one signifies absence by making a (graphic) presence,
by writing something down, inscribing
an unambiguous, permanent, transportable mark.
—Brian Rotman (1987) Signifying Nothing p.59.

I began studying Spencer-Brown's *Laws of Form* (LoF) as a respite from carpentry while building our home in Hawaii in the nineteen seventies. I discovered only later that Charles Sanders Peirce had developed the same iconic boundary approach to logic a hundred years earlier. In the same era Gottlob Frege was the first to formalize predicate logic with his iconic *concept writing*. It seemed both fascinating and outrageous that the formal foundations of rationality were originally iconic, *and* that the opportunity to embody logic within experience and perception was soundly rejected by the academic community.

In the nineteen eighties artificial intelligence began to raise its computational head as *expert systems*. The possibility of learning about intelligent tools for classroom teaching and about new types of intelligence, artificial though they may be, was again fascinating and outrageous. So I went back to grad school at Stanford majoring in Education and AI. There I discovered that AI was just an intense version of symbolic logic. I used LoF as the sounding board for learning symbolic logic and its computational body, and ended up doing an AI dissertation about seventh grade students learning algebra. If you look closely enough, AI expert systems are not able to understand how kids learn, mainly because algebra errors come from outside of algebra, from the context of learning rather than from the content of algebra. It is both wrong and degrading to ascribe biological properties such as awareness, motive, emotion and situated intelligence to computational systems.

I also met mathematics Professor Louis Kauffman in the eighties. He inspired me to explore rigor creatively; his influence permeates the content of this volume. Several scholars have studied LoF within their chosen disciplines. Louis Kauffman developed mathematical extensions of Spencer-Brown's laws of boundary logic, Francisco Varela modified LoF to model the processes of biological autonomy, Heinz von Foerster applied LoF to cybernetics, Frederick Von Meyer applied it to architecture, Jack Engstrom to consciousness studies, Bernie Lewin to ancient philosophy, Niklas Luhmann to sociological analysis, Leon Conrad to narrative and creative writing. Graham Ellsbury, Leon and several others studied under Spencer-Brown and were instrumental in bringing the Spencer-Brown Society into existence to continue his legacy. Don Roberts and Jay Zeman explained to the public Peirce's invention of boundary logic, his *Existential Graphs*. John Sowa, A-V Pietarinen, Sun-Joo Shin and Frithjof Dau studied and refined Peirce's original work. All of these folks who shared their own experiences and expertise thinking about and using boundary logic have helped me to understand it. However, my primary teacher was implementing LoF in dozens of different computational languages. Those experiences led me to strongly diverge from other perspectives, mainly due to the details necessary to make boundary logic useful for computation.

Well, I graduated and got a good job in an AI research company armed with two completely different ways to understand what artificial intelligence was, one symbolic and the other iconic. I applied the same contrasting logical perspectives a few years later within virtual reality systems, and a few years after that within programming languages and silicon architectures. Much of the content in this volume was developed around the turn of our current century. I was working at Interval Research Corporation, Paul Allen's Silicon Valley research lab, with my long-time intellectual partner, Richard Shoup, and a team of inventive computer scientists including Jeffrey James, Tom Etter, Fred Furtek and Andrew Singer. Dick Shoup's *Natural Computing Project* was tasked with this challenge: if you could redesign computing from the ground up, without any consideration for what already exists, what would you build? Dick and I had studied LoF together. We understood the iconic boundary forms not as philosophy but as tools for designing and implementing better software languages and for building better computational hardware. We used LoF as the foundation upon which to modernize silicon computation. Volume II is intended to share the details of the technical work that grew out of the Natural Computing Project.

The content of this volume is limited to propositional logic, one of the simplest systems in mathematics. Timed Boolean logic is also at the foundation of silicon computation. The development of unary logic was guided neither by mathematical proof nor by philosophical clarity but rather by software and silicon pragmatism. Both symbolic arithmetic and symbolic logic are so deeply ingrained within our culture and within our educational systems that it would be foolish to suggest that the symbolic way we understand calculation and rationality should change. But with the advent of the personal computer, symbolic processing itself has changed fundamentally.

In middle school in Australia I was taught (and tested upon) how to calculate the square root of a number by hand. Today smart phones have obsoleted doing any arithmetic by hand. In contrast, natural deduction is still entangled within the complexity, redundancy and ambiguity of language. There is still a mysterious assumed correlation between behaving rationally, solving for unknowns in high school algebra, and proving theorems in high school geometry. Whole number arithmetic teaches us how to see the world as finite countable collections of discrete objects. Natural deduction teaches us to see the world in terms of dichotomous opposites. The middle ground of palatable contradiction is excluded; the only bridge between `TRUE` and `FALSE` is negation. Symbolic logic structures our computational algorithms and is also supposedly better for decision-making in our brains. It is also a *cultural choice*. There are more elegant, more realistic and more humane options.

Clear thinking does not require the comfort, the confusion or the rigid blindness of `TRUE` and `FALSE` and `NOT` and `IF`. Truth is not absolute, it is not even necessary. Negation and implication do not exist in physical reality. Unary logic escapes logical duality by focusing on *contextual relevance*. Deduction requires nothing more than deleting the irrelevant clutter. This kind of simplicity is outside of our cultural heritage and our educational practices. Unary logic is an entirely different approach to rationality. It is not a different language for conventional logic, it is a different way of thinking about what logic is. The shift is not cosmetic, it is postsymbolic.

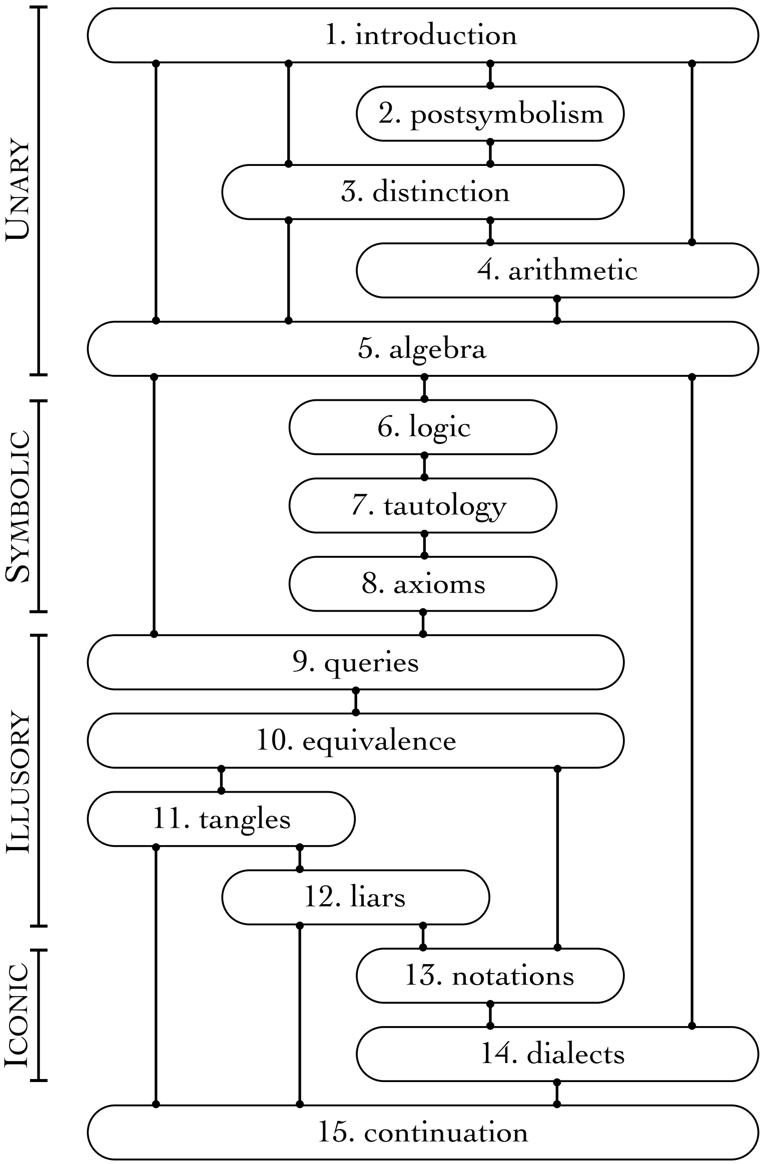
william bricken
Snohomish Washington
June 1, 2026

Global reference:

Page number references to G. Spencer-Brown's *Laws of Form* are for the original 1969 hardback edition published by George Allen and Unwin Ltd. London

References to Charles Sanders Peirce are from *The Collected Papers of Charles Sanders Peirce*. Electronic edition of Vols. I-VI C. Hartshorne and P. Weiss (eds) Harvard University Press, 1931-1935, Vols. VII-VIII A. W. Burks (ed) (1958).

Chapter Map



Left path: illusory queries and examples (1,5,9,10,11,12)
Middle path: unary and symbolic logic (1,2,3,4,5,6,7,8)
Right path: iconic dialects (1,4,5,14)

Guide to Content

This volume defines unary logic and shows how it can be applied to simple deductive problems. Simple problems are primarily for pedagogy however this is not a textbook, there are no problem sets and no exercises for the student. The examples illustrate the structural use of unary logic and hopefully provide guidance for how to think about new logical concepts such as iconic containment, void-equivalent deletion, semipermeable boundaries, illusory forms and interactive dialects, while at the same time tolerating the absence of nearly all of the concepts associated with conventional logic.

Chapter 1 summarizes the conceptual and structural content. It's a potentially useful overview and advanced organizer. There are then four separate sections that explore individual themes in more depth.

UNARY: Chapters 2 and 3 provide quasi-philosophical grounding for the central concepts. Chapters 4 and 5 are quasi-mathematical primers on the structural rules of unary forms. There is no conventional logic in this section.

SYMBOLIC: Chapters 6, 7 and 8 deconstruct propositional logic and extensively illustrate how logic reduces to unary forms.

ILLUSORY: Chapters 9 and 10 introduce querying with illusory forms, a central innovation arising from the void-based perspective established in the Unary Section. Chapters 11 and 12 show the application of illusory queries to different types of deductive problems.

ICONIC: Chapters 13 and 14 describe several varieties of symbolic and iconic representation facilitated by the unary structural rules.

The taste but not the mechanism of unary logic can be experienced by skipping over the explicit demonstrations of computational reduction. Symbolic logic can be avoided by skipping the Symbolic section altogether. If you are interested in iconic forms and representations, both the Symbolic and the Illusory sections can be skipped. Following and doing the algorithmic reductions is a time honored way to learn how to use unary logic as a replacement for what is taught in introductory and intermediate courses on symbolic deduction.

Unary Logic

Volume I

Chapter 1

Introduction

The precise business of logic is to explain the nature of the human mind.[1]
— Encyclopedia Britannica (1771)

CHAPTER CONTENTS

The logic that we learn as two-year-olds, the `NOT`s and `AND`s and `OR`s and `IF`s, comes with a terrible price: *logical dualism*. It teaches us to divide our world into things that are `TRUE` and things that are `FALSE` while eliminating both compromise and contradiction. Unary boundary logic is a quite different way to think about logic. Unary logic abandons both textual symbols and logical dualism [2]. The ground truth of logic is replaced by *potential utility* and *contextual relevance.* The logical value of `FALSE` has no utility and is thus completely eliminated as meaningless. Symbols are replaced by icons that look like what they mean. The result is a **unary formal system** that achieves the *same formal results* as our value-laden logic while eliminating most of the structural rules of symbolic mathematics. Unary logic challenges us to learn a different type of rigorous thinking, one that is much simpler, one that is integrated with our bodies, and one that approaches logic as a unified whole based on the single idea of *containment.* If symbolic logic is a guide to clear thinking, then unary logic provides a different way to understand how to think clearly.

Unary logic is **postsymbolic** and **postdualistic**. It is intended to help to free rationality from tangled streams of written and spoken words and from the fractured perspective of truth-values. This volume compares unary logic to deductive symbolic logic, provides an historical context, and includes many examples of the conceptual and computational benefits of a logic without symbols and without dichotomy. This initial chapter follows closely the introduction of iconic logic by Charles Sanders Peirce's *Existential Graphs*, and George Spencer-Brown's concise mathematical text *Laws of Form*.[3] The chapter provides a foundation for and an overview of *void-based thinking*. It provides immediate exposure to the powerful constraint-based perspective; serves as a ready reference to and a quick reminder of unfamiliar concepts; and emphasizes the fundamental cognitive and computational changes introduced by taking *nothing* seriously. Not only does this perspective provide new tools for thought, it allows us to develop a rationality that is based in physical experience rather than in symbolic abstraction.

deductive logic is an ancient language

Charles Sanders Peirce 1839–1914

George Spencer-Brown 1923–2016

1.1 Logic

Logic is revered as the foundation of rationality, validity and critical thinking. It is a fundamental tool of mathematics and computation. Logic grounded by truth and falsity was introduced by Aristotle 2400 years ago. Western cultural values that are built upon the rule of law and the scientific method have taken logic to be the proper way to use our minds for over two thousand years. Logic is rarely taught as general education in colleges. Two-year-olds learn the words that represent the logical connectives, beginning with `NOT` and `IF`. Adults use these words daily yet we are hard pressed to define them. And since language supports multiple interpretations and conversational ambiguity, our use of the words of logic often does not align with what they are intended to mean. Here we will dispense with them entirely. Try the exercise in the margin.[4]

Aristotle 384–322 BCE

Imagine a parking lot with 100 cars. Estimate in your mind the number of white and blue cars.

Symbolic logic is expressed using typographic symbols such as $\neg$, $\wedge$, $\rightarrow$ and $\forall$. It is a *formal system* with specific definitions and specific rules that must be followed rigorously. **Iconic logic** is a *postsymbolic* formal system expressed using icons and images rather than symbols and text. **Boundary logic** is a type of iconic logic that has only one relationship, *containment*, and only one constant, the *empty container*. **Unary logic** is a void-based boundary logic with structural change based upon *deletion* rather than upon substitution and rearrangement.

Gottlob Frege was the first to formalize predicate logic in 1879. To do so he created an iconic language. Charles Sanders Peirce first introduced container-based boundary logic in the 1880s. George Spencer-Brown extended Peirce's work to include void-based definitions in 1969. However until very recently academic communities have bypassed Frege's and Peirce's and Spencer-Brown's iconic systems in favor of a plethora of words and abstractions.

Gottlob Frege
1848–1925

Controversy

Mathematics can be controversial in many different ways. The most common controversy is not about math but rather about *math education*. Math classes can be brutal. One way that math education challenges young learners is that the foundational rules of how math works are not taught. They are rarely mentioned and never justified. Elementary math textbooks begin in the middle of the story. As an exercise, try to name some fundamental rules that apply to all mathematical systems.

Infinity, ∞, is sometimes controversial. It is an integral part of advanced math that is fairly well understood. Children playing the Name-The-Biggest-Number game quickly arrive at Infinity. And beyond! But sadly infinity is not a number, it's a concept. Herein we will not go anywhere near infinity.

$1/0 = \infty$

Zero, 0, was once highly controversial. It took centuries to gain acceptance and now is commonplace. Unary logic contributes to the *illegitimacy* of `0` by eliminating all representations that indicate nothing. *Void-based* unary logic asserts that *non-existence*, what we might call void, has absolutely no properties. This perspective eliminates the belief that nothing can be indicated by something.

2+3=5 is TRUE

2+3=4 is FALSE
2+3=4 is FALSE
2+3=4 is FALSE
2+3=4 is FALSE
2+3=4 is FALSE
2+3=4 is FALSE

Quite plainly, unary logic is not conventional logic. It does however have the same capabilities as symbolic logic without the mechanisms and ministrations that are taught as logic. The concept of `FALSE` is not implicit, not hidden, not implied. In unary logic the concept simply does not exist. The concept of negation is no longer relevant. It is here that unary logic finds controversy, elegance and new methods of computation. We will be taking a journey into an entirely different kind of rigorous rational thinking, one that embraces the positivity of existence and that reconnects to our experience, to our intuition and to our common sense.

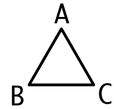
Symbols and Icons

Symbolic logic uses symbols to express the abstract concepts of logic. A **symbol** is a representation that bears no resemblance to its meaning. We use symbols composed of letters, punctuation marks and other characters that fit nicely into lines to provide typographic convenience. We have developed calculi that automate computation by manipulating symbolic strings of zeros and ones. We mix up random symbols and call the resulting mess a password.

Yes, emojis are iconic

Iconic logic uses icons and images to reconceptualize the concepts of conventional logic. An **icon** resembles what it refers to. It is a *visceral form* of meaning that can be sensed and visualized. We express emotional content with emojis. Our passwords are images of fingerprints and faces. Iconic math is already widely taught by elementary school manipulatives such as Cuisenaire rods and by Cartesian graphs in high school algebra. Iconic representations such as geometric figures, family trees, navigational maps, star charts, architectural blueprints, engineering diagrams, ontological graphs, periodic tables and statistical spreadsheets, are fundamental for organizing science and knowledge.

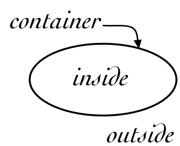
Some mathematical forms, such as an equilateral triangle, $\triangle$, are inherently iconic. An icon can be transcribed into symbols, such as the words "equilateral triangle", or into equations derived from labeling the parts of the triangle, $AB = BC = CA$, or even into chimeric combinations of text and icons, $\angle A = \angle B = \angle C$. The icon in the margin, however, carries more information since it illustrates as well as describes.



Proof of mathematical facts using diagrams and images was professionally banned about two hundred years ago when mathematicians were making too many mistakes using images to support abstract reasoning. Calculating with icons was believed to be difficult, clumsy and inherently error-prone. Symbolic text however is tied to the media environments of a century ago. Displaying and calculating with icons is no longer difficult. The fundamental presumption that math is expressed using strings of symbols is beginning to fade.

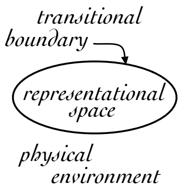
1.2 Building a Boundary Language

We begin by following the common sense consequences of a very simple idea. We'll construct an iconic system based on *containment*. A **container** has an *inside* and an *outside*. Containers have only one function, they contain things on their inside. *Example*: Most of the objects in a kitchen are containers. Boxes and cans contain food. Cupboards and shelves contain these containers of food. The walls of the kitchen contain all of the above. We'll focus on the edges of a container to construct a *logic of boundaries*.



Clear a Space

One of the first things we do when communicating formally is to clear a space for writing and drawing. The cleared space rests upon a physical substrate like an empty sheet of paper or a computer screen when the monitor is unplugged. The only role that the cleared space plays in the meaning of letters and words is to allow them to look different. In contrast the background can contribute dynamically to the meaning and simplification of iconic forms.



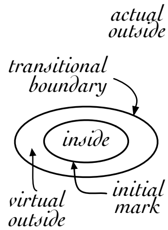
The **transitional boundar**y of the physical substrate delineates a *virtual representational space*. Before it is marked upon, the representational space is *void* or more simply, empty. I'll use a smaller font size for the word *void* to emphasize that the **representational space** has absolutely no structure, no properties and no characteristics. Emptiness is ineffable, without distinction or indication of any kind. When we refer to the nothing within an empty container, we are referring to *a property of the container*.

Non-representation cannot have meaning but absence can have utility. Void-based logic does not need to include the linear concepts of ordering, grouping and the number of inputs to a function (commutativity, associativity and arity respectively). Instead bounded forms are independent of one another and constrained only by the edges of their outer container. Void-based deduction proceeds by *deleting what is irrelevant* rather than by accumulating what is factual. The approach is *constraint-based* rather than goal-directed. Constraint-based deduction optimizes rather than evaluates, leaving all possible conclusions rather than identifying one conclusion as TRUE.

Containment

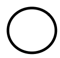
The transitional boundary separates the *physical* (the writer, the paper page, the pen) from the *virtual* (the symbolic and iconic representations therein). The physical is outside, the virtual is inside, and the physical boundary **contains** all virtual representations. All virtual structure is constructed within and thus contained by the boundary of the space of representation.

The initial act of representation is to make some sort of mark within the substrate. To maintain simplicity, we'll adopt a discipline to *introduce no additional concepts other than that of containment*. No words, no numbers, no shapes, no metrics, no logics, no geometries, nothing else. The transitional boundary of a representational space is the *only* thing available to represent. Rather than invent some new squiggle that represents the boundary, we can maintain simplicity by making a mark that looks like the transitional boundary. We'll call the *iconic representation* of the transitional boundary an **initial mark**. The mark then is also a container. The difference is that it is entirely virtual, we no longer need to consider the physical substrate.



This Page

I'll reserve the margins that bound these words to show rather than to tell. In the margin two-dimensional containers are illustrated by closed circles and rectangles. In the text *delimiter pairs* (parentheses, brackets, braces, quotation marks, etc.) represent containers. I'll call parentheses that represent formal containers **parens**, `( )`, and depict parens in Monaco font. Parens have no "left-side" or "right-side", they are a linear approximation of a contiguous enclosure.



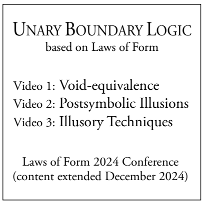
`(` *inside* `)` *outside*

Iconic math is **postsymbolic**. It enlists images and experiences to dissolve the syntax/semantics barrier of symbolic notation. To soften the irony of overusing words, here are links to three narrated videos with images and animations of unary logic.

Unary Boundary Logic Part I: Void-equivalence
`https://vimeo.com/1040314440` (45 minutes)

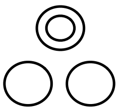
Unary Boundary Logic Part II: Postsymbolic Illusions
`https://vimeo.com/1040314031` (21 minutes)

Unary Boundary Logic Part III: Illusory Techniques
`https://vimeo.com/1039534775` (41 minutes)



Replicas

Without adding new concepts, we can draw another mark within the representational space. *Marking again* allows us to construct configurations of virtual boundaries using **replicas** of the initial mark. Each act of replication comes with a choice, since we can place the replica on the inside or the outside of an existing mark. *Example:* In what ways can two boxes be arranged? They can be outside of each other, or one can be inside the other. This choice can be visualized both with parens and with spatial enclosures such as circles:



`(())`	a mark inside another mark
`()()`	two marks outside of each other
`(()())`	two marks within the same outer container

The configuration with exactly one container inside another, `(())`, occurs frequently. We'll call it a **nested-pair**.

textual examples of forms

`((((( )))))`

`(( )( )( )( ))`

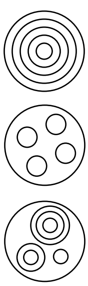
`(( ) (( )) ((( ))))`

The second configuration, `()()`, with two marks outside one another is *not* a containment relation and thus not part of the boundary language. This configuration can however be contained by an outer container, `(()())`. Every configuration of boundaries is constrained to have a *single outermost container*. Any configuration of containers bounded by an outermost container is a **form**. All forms thus represent containment relations. A **containment relation** is between a containing mark and a contained mark. This is the *only* structural concept in unary logic.

Constant

Forms are changed by transformation rules. The *only* form that cannot be simplified is the empty container.

iconic examples of forms

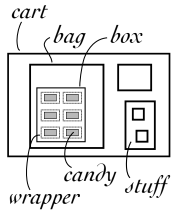


`( )`	*constant*

The empty container is a **mark** that represents a boundary. It's the **ground** upon which unary logic rests. It's the only **constant** form. And it's a **frame** around *void*. I'll reserve the word *mark* to identify the boundary constant.

Forms

The *language of boundary forms* is all the different ways containers can be nested without overlap. I'll use small letters {`a`, `b`, `c`, ...} as labels for individual forms. Labels are not part of the boundary language, we need them only as a convenience to mutually identify forms. A concrete example of a form is several candies each inside its own wrapper all inside a box inside a shopping bag inside a cart. The margin shows an image of this containment form. The image also shows some *incidental details* that are not relevant to forms. We do not have, and will not need to introduce, the concepts of size or shape or number or distance or proximity or orientation. Here's the cart with candy expressed as a parens form:




```
(   (   ( (()) (()) (()) (()) (()) (()) )   ) ( ) (( )( ))   )
|   |   |      |          |                   |      |
cart bag box   wrap       candy               other  stuff
```

When a form contains multiple other forms, those other forms are called **subforms**. *Example:* The wrapped candies are subforms of the candy box. The bag and the other stuff are subforms of the cart.

Transformation Rules

Transformation rules allow us to *construct* and to *simplify* forms. *Example:* We simplify a messy numeric expression by making it equal to something simpler. I'll use an equal sign, =, to signify changes that are supported by rules. This is not the same equal sign as what is taught in school arithmetic. We have no concept of numeric value or keeping values equal. The boundary equal sign allows us to assert that the mark is unique and unambiguous. There is no other form that looks like it.

1+1+3+2+1 = 8

$$\boxed{(\ ) = (\ )}$$

identity

The constancy of the constant is called **identity**. Identity generalizes to every form. Unlike symbols, *every form looks like what it is*. The equal sign also asserts **exceptions** in which two forms that appear to be structurally different are taken to be functionally and pragmatically the same. We would say they are *equivalent* even though their subforms may differ.

identities

(( )) = (( ))

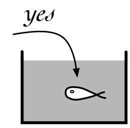
(( )( )) = (( )( ))

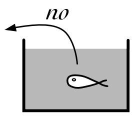
equivalencies

((( ))) = ( )

(( )( )) = (( ))

Transformation rules define equivalent containment patterns that can be substituted for one another. Which patterns are equivalent is mostly an arbitrary choice depending on what we want to hold constant across the substitutions. Symbolic logic employs equivalence rules that maintain truth-values. Void-based unary rules identify meaningless or redundant or contextually irrelevant forms



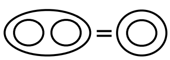


that can be freely deleted without impacting equivalence. *Example:* Rules for a home aquarium as a fish container: Put fish into the tank. Don't let fish out. The name of a fish is not relevant to keeping the fish in the tank, so saying "Put Sally-the-Guppy fish into the tank" is equivalent to saying "Put the fish into the tank". For the purpose of keeping the fish alive, its name is contextually irrelevant.

Calling

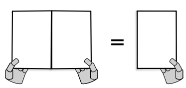
To simplify forms we'll need only two rules, one to reduce (( )( )) and one to reduce (( )). The **Rule of Calling** eliminates replica marks within the same container as redundant and unnecessary. It also makes counting impossible.

to call



(( )( )) = (( ))   *delete replicas*

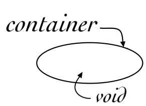
Example: Two nickels equal one dime. If you have two nickels in your pocket, you can exchange the two nickels for a dime while keeping your wealth constant, and you will have less stuff in your pocket. *Other examples:* Two empty pages provide the same functionality as one. Two pens permit marking to the same extent as one pen.



It may be desirable to change from simple to complex. Transformation rules based on equivalence can be applied in both directions. *Examples:* We may have a dime but be facing a gumball machine that takes only nickels. We may want to have a spare pen just in case or we may want to write more than will fit on a single page.

Crossing

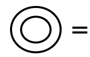
The **Rule of Crossing** simplifies nested-pairs. But there are no forms that are simpler than a nested-pair (( )) or a mark ( ). Here is where unary logic exhibits its power. We have another option for simplification: *non-existence*. Both Peirce and Spencer-Brown introduce the structural idea that some forms can be both created from *void* and deleted into *void*.[5] The idea of non-existence has been within the container system all along, as the interior of the initial mark. Before that, the unmarked representational space is itself *void*.



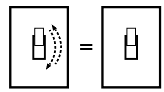
Crossing is void-based. It defines a nested-pair to be *void-equivalent*. A nested-pair is the same as no form at all.

(( )) = *void* *delete nested-pairs*

to cross

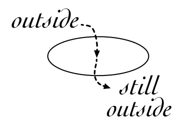


We are free to delete nested-pairs wherever they occur. We can also create nested-pairs anywhere there is empty space. And *empty space is everywhere* since the substrate pervades all forms. The symbol *void* on the right-side of Crossing is a reminder that the form on the left-side is specifically equivalent to its own absence. Crossing asserts that the nested-pair changes nothing, it is irrelevant. *Example:* The purpose of a light switch is to change the illumination of a room. Flipping the position of the switch once will change the state of a light. Flipping it again will return the light to its original state, as if nothing has changed.



light	*form*
off	*void*
on	( )
off	(( ))

Calling treats the container as an **object**. It asserts that replicated marks in the same container are redundant. Crossing treats the container as a **process**, as the activity of containing. It asserts that repeating the action of containment reverses the original containment. Crossing the boundary of a container changes our perspective from one side to the other. By crossing again, we return to the side that we started from. The pair of crossings become irrelevant, our location has not changed. *Examples:* Put an apple into the basket and then take the apple out of the basket. The relationship between apple and basket has not changed. Open the door and the cat comes in. Open the door again and the cat goes out. The location of the cat has not changed, it began outside and it ends up outside.

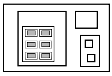


1.3 An Arithmetic of Boundaries

An *arithmetic* is a formal system of rules for combining specific objects. Those objects are almost always taken to be numbers. Here the objects are containers.[6] These are the pattern transformation rules for combining containers:

(( )) = *void* **to cross**

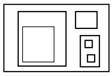
(( )( )) = (( )) **to call**



Let's try these two arithmetic rules on the cart with candy:

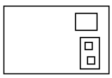
((((()) (()) (()) (()) (()) (()))) () (()()))

cart bag box wrap candy other stuff



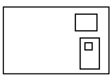
We'll first unwrap and eat the candy by applying Crossing to each wrapped candy:

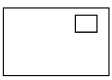
((( )) () (()()))



Now the candy box and the bag it is in can be discarded, again using Crossing:

(() (()()))

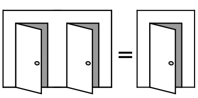


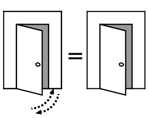


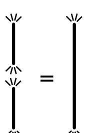
All that's left is the cart and the other stuff. We can reduce the box that contains replicas of other stuff using Calling and then continue by using Crossing to remove that box entirely. What remains is the cart and one empty container:

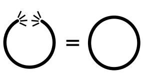
(())

If we apply Crossing once again, both get discarded and we are left with nothing at all. In other words, we no longer need the cart. The transformation rules work in both directions. We can get a shopping cart and put some stuff in it, and then we can put the box of wrapped candies into a bag and put the bag into the cart.









There are many other simple examples. Consider doors in a room and their *purpose* for entering and exiting the room. The parens ( ) is now a door. The wall is the outermost frame. *Calling:* replica doors are equivalent to one door. *Crossing:* closing the door and then opening it again results in no change. Consider parens now as locks for the purpose of rendering the door unusable. Replica locks are equivalent to one lock. Locking and then unlocking a door results in no change, the door remains unlocked. Consider the ends of a piece of string. *Calling:* two separate pieces join into one piece by tying two of the ends together. *Crossing:* tying the two ends of a single piece of string results in no ends. In the other direction cutting a piece of string adds two new ends whether or not the string is in a loop. The number of pieces however varies.

Modifications to *Laws of Form*

Spencer-Brown modernized Peirce's Existential Graphs. Unary logic modernizes *Laws of Form.* These modifications are implicitly recognized within Spencer-Brown's text. Here they are central.

Singular Forms

All forms are *singular* containers with arbitrary contents. The single outermost container allows all forms to be mutually *independent.* Each form is in a containment relation with its container and is not related to other forms in the same container.

Void-based Rules

Both of the transformation rules for unary arithmetic are void-equivalent since Crossing reduces the right-side of Calling to *void*. This convergence makes unary arithmetic purely void-based. The rules define some forms to be *meaningless* and to exist only as *transient illusions*. Void-based Calling has the same intent as Spencer-Brown's unbounded Calling. I'll use the single name Calling in this volume for the version with an outermost boundary. Throughout the text whenever a transformation rule is applied, I'll make note of the transformation in the margin.

(()()) = (()) = *void*

(( )) = *void* `cross`

(( )( )) = (( )) = *void* `call, cross`

The constant mark is not meaningless. For example:[7]

(((   ))) = ( ) `cross`

((( )( ))) = ( ) `call, cross`

Emphasis on Parallelism

Spencer-Brown defines computational steps in a classical manner by making changes one at a time. Due to their independence, unary forms are not limited to sequential one-at-a-time processes. All transformations are *naturally concurrent*. *Example:* We can take all of the wrapped candies out of their box at the same time.

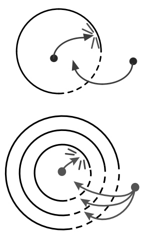
parallel calling (( )( )( )( )( )) = *void*

parallel crossing (((( )) (( )))) = *void*

Generalized parallel Calling will still be called Calling.

Semipermeability of Boundaries

semipermeability



The most important modification is that containers are *semipermeable from the outside*. A logic boundary constrains only what is on its inside. Semipermeability makes all inward facing boundaries non-existent to forms on their outside. A pair of *vertical ellipses*, ⋮ ⋮, indicate semipermeable boundaries in rules.[8] They are intended to look like porous parens. *Example:* We can put groceries into the cart without changing our wealth, but when the cashier takes them out our wealth decreases.

1.4 Unary Boundary Algebra

An *algebra* of containers needs only one additional idea, a **pattern-variable** to stand in place of *all possible forms*. I'll use capital letters {A, B, C, ...} as labels for pattern-variables. Unary boundary algebra consists of three void-based rules that identify patterns that can be deleted and created.

Occlusion

occlude (A ( )) = *void*

Occlusion generalizes Calling. Any mark deletes the container it is in together with all of the contents of that container. *Example:* Consider the outer parens to be a table. The mark is an empty cup of coffee on the table. The pattern-variable A represents everything else on the table. For the purpose of drinking coffee, the empty cup makes everything on the table (and the table itself) irrelevant.

Involution

involute ((A)) = A

Involution generalizes Crossing. We can delete nested-pairs regardless of their contents. We can also construct a nested-pair anywhere within the representational space. *Example:* Consider the outer parens to be a pocket and the inner parens to be a coin purse. For the purpose of buying a cup of coffee, the pocket and the purse are irrelevant. All that matters is the money nested within the two containers. We can also put the money into the purse and the purse into a pocket without devaluing our money. Involution provides **grouping by containment** without the need for additional concepts or notations. A direct consequence is that any collection of forms can be considered to be singular, with the outermost boundary provided by a nested-pair.

involutions

```
void  = ((     ))
A     = ((  A  ))
A B C = ((A B C))
```

Pervasion

pervade

```
(A ⁝A⁝) = (A ⁝ ⁝)
```

Pervasion defines boundary *semipermeability*. It generalizes Calling to any forms and across all depths of nesting. Any forms existing on the outside of a boundary can be arbitrarily deleted from or inserted into the inside at any depth. *Example:* The whitespace of a sheet of paper allows us to replicate our thoughts anywhere on it. An outermost form is the **host** of its nested **replicas**. The **vertical ellipsis**, ⁝ ⁝, indicates that any boundaries between the host `A` and the inner replica `A` are transparent. The margin shows some reductions based on the semipermeability of boundaries. Succinctly, *pervaded replicas are void-equivalent*.

pervasion

```
A (A)
A ( )

A (B (A C D))
A (B (  C D))

A (B (C (A B C D)))
A (B (C (      D)))
```

valid pervasion

```
A ( ) ≠   (A)
A ( ) = A (A)
A (B) ≠   (A B)
A (B) = A (A B)
  (A) ≠ A (A)
```

Transformation

Unary logic transformations are *independent* of the *context* in which they are applied. Forms other than the pattern `A` are not mentioned in the rules and can be ignored when applying the rules. In Figure 1-1 the three rules have graduated to being called **pattern axioms**. These axioms define how the unary system works. They are accepted without proof. Each axiom identifies an optional deletion pattern when read left-to-right and an optional construction pattern when read from right-to-left. **Logical deduction** is thus defined as *deletion* and *creation*

Axioms of Unary Logic

Ground Pattern Axioms

```
(( )) = void           crossing
(( )( )) = void        void calling
```

Algebraic Pattern Axioms

```
(A ( )) = void         occlusion
(( A )) = A            involution
(A ⁝A⁝) = (A ⁝  ⁝)     pervasion
```

Figure 1-1: *Summary of Unary Pattern Axioms*

of meaningless, redundant and contextually irrelevant forms. It's almost always a good idea to delete irrelevant forms immediately. The axioms are independent so we don't need to think about which one to apply first. For any reduction all we need to do is to identify one relatively simple specific pattern. If there is a ground, delete it by Occlusion. If there is a nested-pair, delete it by Involution. If there's a host form in a container then delete any replicas at any depth within that container using Pervasion. Here is an example of each pattern rule in action.

occlude	*involute*	*pervade*
`(A (B ( )))`	`(A ((B C)))`	`(A (A B C))`
`(A         )`	`(A    B C   )`	`(A (   B C))`

Here's an example of reducing a form that uses all of the rules, with further reductions facilitated by prior deletions. Changes are visible line-by-line as the deletion of subforms directly above.

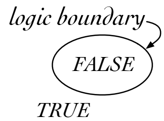
```
               (a (b (b (a c)) (a ((c)(d))) ))
 pervade b     (a (b (  (a c)) (a ((c)(d))) ))
 pervade a     (a (b (  (  c)) (  ((c)(d))) ))
  involute     (a (b       c       (c)(d)   ))
 pervade c     (a (b       c       ( )(d)   ))
   occlude     (a                            )
```

By reading upwards, we can see a demonstration of how the initial form at the top can be *constructed* from `(a)`.

1.5 Interpretation as Logic

Here is the remarkable feature that both Peirce and Spencer-Brown discovered. The *axioms of unary logic* are sufficient to replace a *small but fundamental branch* of traditional logic, **propositional reasoning**. Unary logic makes their discovery computationally elegant.

In place of `TRUE` and `FALSE` unary logic uses the two different spaces (outside and inside) created by the mark. The outside is `TRUE` and the inside is `FALSE`.



( )	`TRUE` ground
void	`FALSE` does not exist

Within unary logic `FALSE` is an *illusory concept*. Forms that can be interpreted as `FALSE`, regardless of complexity, are *meaningless*. **Illusory forms** can be brought into apparent existence whenever we like wherever we like and then returned to non-existence when they are no longer useful. Chapters 9 through 12 provide many examples.

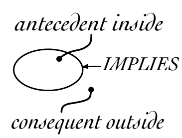
Logical Connectives

Figure 1-2 provides the map to express the connectives of conventional logic as boundary forms. Capital letters stand in place of any logic formula. The map is *many-to-one*, many different logic formulas transcribe into the same boundary form.

A boundary can be *interpreted* as the logical negation of its contents. A mark negates its interior *void*. Thus ( ) can be read equally as `TRUE` or as `NOT FALSE`.

( ) `TRUE = NOT FALSE`

The inside of a boundary *implies* the outside, providing another reading of the mark.



( ) `FALSE IMPLIES FALSE`

Void can also be read as disjunction, `A OR B`. This provides other readings of the mark:

( ) `TRUE OR FALSE`
( ) `FALSE OR (FALSE IMPLIES (FALSE OR FALSE))`

symbolic logic ☞	*unary logic*
TRUE	( )
FALSE	*void*
A	A
NOT A	(A)
A OR B	A B
A IMPLIES B	(A) B
A AND B	((A)(B))
A IFF B	((A)(B))(A B)

Figure 1-2: *Map from Symbolic Logic to Unary Logic*

Encoding

Here is an example of the **encoding** process, translating propositional logic into parens, the deepest connective first. The transcription icon ☞ is described in Chapter 2.4.

```
              a AND NOT (b OR c)
OR                      (b OR c)  ☞        b c
NOT                 NOT (b OR c)  ☞       (b c)
AND           a AND NOT (b OR c)  ☞  ((a) ((b c)))
```

The inherent redundancy in the logical connectives often encodes as a nested-pair that can be immediately deleted by Involution.

```
involute      a AND NOT (b OR c)  ☞  ((a)   b c   )
```

Decoding

Due to the one-to-many mapping, how a parens form is read as logic is at the discretion of the reader. For example, the form ((a)(b)) can be read as a variety of logic formulas.

```
((a)(b))  ☞
     a AND b
     (a AND b) OR FALSE
     NOT (a IMPLIES NOT b)
     NOT (b IMPLIES NOT a)
     NOT (NOT a OR NOT b)
((a IMPLIES FALSE) OR (b IMPLIES FALSE)) IMPLIES FALSE
```

How logical computation works using boundaries is the content of the rest of the volume. For now I'll show just one simple example of *void-based deduction*.

Application

The primary inference rule of symbolic deduction is **detachment**: If proposition P is TRUE, and if P IMPLIES Q, then proposition Q is TRUE. Logical equivalence is conventionally represented by a three-bar equal sign, ≡.

```
      (P AND (P IMPLIES Q)) IMPLIES Q      ≡ TRUE
```

We'll first transcribe this expression one logic operator at a time, deepest first:

```
       P AND  P IMPLIES Q   IMPLIES Q      ≡ TRUE
       P AND    (P) Q       IMPLIES Q      ≡ TRUE     IMPLIES
     ((P)      ((P) Q))     IMPLIES Q      ≡ TRUE     AND
   ( ((P)      ((P) Q)) )           Q      ≡ TRUE     IMPLIES
   ( ((P)      ((P) Q)) )           Q      =  ( )     TRUE
```

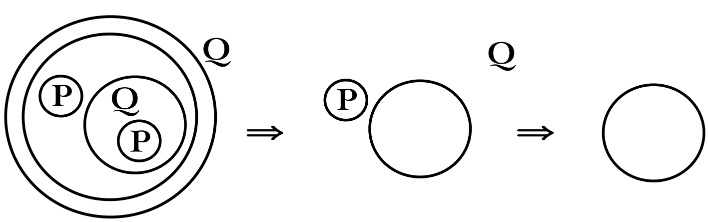
Here's a demonstration of the validity of detachment using unary logic:

```
((  (P) ( (P) Q ) )) Q        =  ( )     transcribe
    (P) ( (P) Q )    Q        =  ( )     involute
    (P) ( (P)   )    Q        =  ( )     pervade Q
    (P) (       )    Q        =  ( )     pervade (P)
((  (P) (       )    Q ))     =  ( )     involute
(                       )     =  ( )     occlude
```

This demonstration is not characteristic of boundary deduction since each step has been shown separately and sequentially. Here is the equivalent parallel boundary reduction:

```
((  (P) ((P) Q) )) Q          form
    (P) (      )   Q          inv, pervade (P) Q
        (      )              occlude (P) Q
```

Here's the same demonstration using Peirce's enclosing circles:



Concept

This volume provides an introduction to void-based thinking but it is not intended to be an instruction manual. Instead there are many examples of the application of void-based computational techniques to deductive problems. The multiple logical interpretations of a form, the absence of both `FALSE` and the concept of negation, the contextual irrelevance of the concept `TRUE`, the absence of both the logical connectives and the rules of deduction all indicate that we are no longer using the *conceptual foundation* of symbolic logic. We have not simply changed notations, we are using a *completely different conceptual perspective*. Rationality becomes deleting irrelevant containers (Occlusion, Involution) and deleting pervaded replicas (Pervasion). No other conceptual tools are needed.

1.6 Coming Attractions

We have redefined both the **formal structure** of the constants and connectives of logic and the **formal transformations** used in deduction and inference. **Inference** is drawing logical conclusions by reasoning from evidence. **Deduction** is principled construction of new facts from established and secure facts. Our task now is to understand what being *logical* might mean when the common concepts of logic that have been with us for millennia are cast not only into an unfamiliar language but also into an unfamiliar conceptual system that uses a non-textual medium of communication. Standing closer, we'll see that unary deduction can be expressed by three simple permissions to *delete* and *create* form. Standing back, we'll see that the structure of logical deduction is incredibly simple.

Chapters 2 and 3 provide a postsymbolic conceptual underpinning and define the cognitive act of distinction. Chapters 4 and 5 describe the arithmetic and algebra of unary logic. Chapters 6, 7 and 8 compare symbolic and unary logic systems. Chapters 9 and 10 introduce illusory form as a tool for deduction and proof. *Illusory forms* support an entirely new proof procedure. They are apparent forms that do not exist, yet they can be brought into temporary virtual existence to catalyze simplification by deletion. Postulating illusory forms by imagining them is not the same as testing a logical hypothesis since there is no specific hypothesis being tested. It is more like asking if there are any subforms that are irrelevant within the context of an illusory form. Chapters 11 and 12 demonstrate the capabilities of illusory forms for solving well-known challenging deductive problems. Chapter 13 connects the axioms of unary logic to conventional functional and relational perspectives. Chapter 14 presents several *interactive iconic dialects*. A second Volume II is projected to describe some technical applications of parens forms and distinction networks, particularly in the design of programming languages and semiconductor circuitry.

1.7 Remarks

Unary logic provides an unusual evolutionary perspective by viewing historical European intellectual reliance on linguistic redundancy and on symbolic duality as a *cultural choice* made at the turn of the twentieth century when the iconic tools of the founders of formal logic were passed over in favor of symbolic representation. At the time that decision was not entirely irrational since pixel-oriented displays did not yet exist and rationality was explored and debated solely within the medium of spoken words and written symbols. *The medium was indeed the message*. Being logical meant following chains of implications, a technique inherited from two thousand years of syllogisms. More generally the founders of symbolic logic were world-leading mathematicians interested in creating a firm foundation for all of mathematics. Their ambitions obscured the possibility that everyone might benefit from a simpler understanding of logic and rationality. In the age of digital convergence unary logic provides an experiential alternative, just as reasonable as symbolic logic but with much greater respect for Ockham's (and Einstein's) recommendation to *keep it as simple as possible*.

William of Ockham
c.1280– c.1349

Albert Einstein
1879–1955

Endnotes

1. **opening quote:** *Encyclopedia Britannica* (1771) Volume II p. 984.

Here's the entire first paragraph under *Logic:* "Logic, the art of thinking and reasoning justly; or, it may be defined the science or history of the human mind, inasmuch as it traces the progress of our knowledge from our first and most simple through all their different combinations, conceptions, and all those numerous deductions that result from variously comparing them one with another." The next paragraph continues: "The precise business of logic, therefore, is to explain the nature of the human mind, and the proper manner of conducting its several powers, in order to the attainment of truth and knowledge."

2. **symbols and logical dualism:** This volume does not address the much broader debate over *philosophical dualism*. There is no mention of the mind/body problem, no attempt to face the difficulties of integrating consciousness with physicality, and no questioning the formal basis of mathematics.

3. **concise mathematical text *Laws of Form*:** G. Spencer-Brown (1969) *Laws of Form*. First Ed. London: Allen & Unwin.

4. **exercise in the margin:** There are most probably zero white and blue cars. The conjunction `AND` connects properties of an individual car. A particular car would need to be both white *and* blue. We tend to read the phrase "white and blue cars" as "white cars and blue cars". The exercise is ambiguous due to conversational abbreviations in our language. A *formal* interpretation, by design, cannot be ambiguous. It is the disjunction `OR` that puts cars that are white together with cars that are blue as "white or blue cars". Our use of language is not *wrong*, it is just more flexible than the formal connectives of conventional logic.

5. **and deleted into *void*:** The null operation is common in mathematics but it is not usually considered to be a feature. We can add zero to any number, `n` → `n+0`, and if zero is being added we can delete the addition, `n+0` → `n`.

6. **objects are containers:** An arithmetic is not always about numbers. Matrix arithmetic is the rules for combining matrices. Boolean arithmetic is the rules for combining bits (zeros and ones). Logical arithmetic is the rules for combining truth-values. A *boundary arithmetic* defines how boundaries are simplified.

7. **not meaningless. For example:** In *Laws of Form*, Spencer-Brown defines Calling without the outermost container, as `( )( ) = ( )`.

8. **boundaries in rules:** I've experimented with many ways to represent the semipermeability of Pervasion. Symbolic notation is very poorly adapted for the expression of nested forms. Chapter 2, endnote 51 has typographic information.

Chapter 2

Postsymbolism

CHAPTER CONTENTS

...in a book we have to use words and other symbols in an attempt to express what the use of words and other symbols has hitherto obscured[1]
— George Spencer-Brown (1967)

Throughout history logic has been associated with language. How to think clearly was expressed in words, while logic was expressed in symbols that stand in place of words. Attainment of knowledge was thus considered to be a purely cognitive activity. **Symbolic logic** was directly associated with the *proper use* of our minds and was taken to be how the mind works. In 1972 Newell and Simon published their seminal cognitive science text, *Human Problem Solving*. In it they declared that *reasoning* is the ability to formally manipulate symbols.[2]

Allen Newell
1927–1992
Turing Award 1975

Herbert Simon
1916–2001
Nobel laureate 1978

Our cultural and academic commitment to symbol processing is a *design choice* not an inevitability.[3]

It is no longer reasonable to believe that cognitive skill depends upon symbol processing.

Today the currency of cognition has shifted to global information access, shared images, uploaded videos, interactive software applications, computationally generated virtual realities and artificially intelligent agents. We are immersed in multimedia technologies within a postsymbolic world.

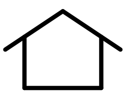
The cultural evolution of logic incorporates *vestigial concepts*, ideas and perspectives that were once useful but have lost their original purpose. We no longer record logic in Greek or in Latin, although both languages were fundamental to the early development of logical concepts. Classical logic embraces the Excluded Middle, propositions are either `TRUE` or `FALSE` and nothing else, an approach that functioned to keep moral and intellectual order prior to the European Renaissance. In this age of contextual truth, we now have modal (possibility and necessity), temporal (was, is and will be), deontic (permissibility and obligation), fuzzy (uncertainty and imprecision), second order (relations as variables), mathematical (models, proofs and computability) and many other types of logic.

LEVELS OF ABSTRACTION

symbol

house

icon



clipart

image

The symbolic perspective continues to confuse intelligence with facile manipulation of meaningless patterns of textual squiggles that are presumed to encode meaning. The capability of artificial intelligence systems to search through and organize massive collections of symbols is extremely useful and certainly beyond the capabilities of humans. But this no more defines intelligence than does the unimaginable complexity of electronic signals streaming in parallel across a labyrinth of tiny wires during silicon computation.[4]

Jaron Lanier 1960–

The perspective of unary logic is that both symbols and duality are vestigial. Learning to abandon the conceptual and notational mire that we have been taught as the rules of clear thinking no doubt will be challenging. A dominant obstacle is the natural propensity to impose previously learned symbolic complexity upon the simplicity of boundary forms. We'll need to see duality as a philosophical debate that does not necessarily depend upon the *constants* or the *structure* of symbolic deduction.[5] Logic does not necessarily need to be about truth and falsity, it can be about relevance. It can be about our physical experience, it can be about simplicity. We'll need to practice what Jaron Lanier calls *postsymbolic communication*.[6]

2.1 Symbolism

A **symbol** is an encoded chunk of potential information. Symbols are communication tools that have an *arbitrary* representation. Words are the symbolic forms we are most familiar with. They specialize in abstract concepts like freedom, humor and compassion. Physicist Julian Barbour: "A symbol can stand for *anything*, but it must stand for *something*."[7] Symbolic encoding is easily standardized. Words however impose a substantial **cognitive load**. The *meaning* of words is multifaceted and highly contextualized. We have to memorize and to recall in order to recognize the patterns that they weave. We *teach* children how to read but not how to walk, talk, see or think. Symbolic abstraction forces our perception to abandon immediate experience in favor of the shadows of our linguistic creations. Dancing, singing, gardening and sports, in contrast, are understood and remembered through direct experience within a physical environment.[8]

Symbolic Diversity

A triangle is a closed geometric figure that, when drawn on a page, has both width and height and thus identifies an area. A triangle has both an *inside* and an *outside*. The symbolic word "triangle" has a length but not an area. It has an outside but not an inside. The inside of a triangle can contain other things, even a word or a number. The characters in the word "triangle" cannot contain other characters.

The meaning of text requires consensual agreement upon definitions, followed by mass memorization. The textual word forces us to think abstractly, while the geometric triangle allows us direct visual and even physical understanding. Words engender an out-of-body experience, while icons and images invite participation. Words are supported by several sensory modalities, we can say and hear "triangle". But the sound of a word, like the typographic representation of a word, rarely contributes to our understanding of its encoded meaning. Vocal inflections and songs, in contrast, are auditory iconics.

triangulus
huinakolu
三角形
segi tiga
driehoek
त्रिकोण
треугольник
სამკუთხედი
trekant
مثلث
unxantathu
Triangle
משולש
háromszög
τρίγωνο
triángulo
త్రిభుజం
трикутник
triangel
гурвалжин
ትሪያንግል
Dreieck
ত্রিভুজ
triongl
ത്രികോണം
Þríhyrningur
삼각형
saddex xagal
ತ್ರಿಕೋನ
өчпочмак
trójkąt
ត្រីកោណ
trekëndëshi
ئۈچبۇلۇڭ
üçgen
တြိဂံ

The margin shows several *textual* forms of the word "triangle".[9] Although they are often beautiful, they are far too diverse to suggest a common referent. The spatial form of a triangle is *iconic*, it resembles what it indicates. If a child is asked to tell us about a triangle, she would describe in words the visual form, or perhaps draw in space or on paper the three straight lines that bound an area. In the margin iconic forms of △ visually assert their common intention. They share a perceptual pattern that makes consensual understanding easy. A Platonic world of ideals is unnecessary. The concept of *triangle* need not be of a perfect form. The iconic compromise is to allow a *generic* shape to stand in place of a generic idea. The iconic triangle need not be equilateral. It could be isosceles, or contain a right angle, or be scalene with all sides of different length. It could even be obtuse.

Symbolic Logic

Propositional logic is *sentential*, a textual style that combines sentences (propositions) using logical connectives to form formulas that are either `TRUE` or `FALSE`. Here's computer scientist Mark Greaves:

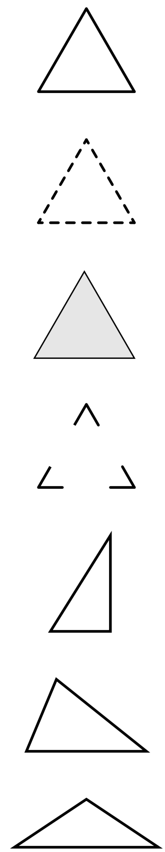
> For a particular set of theories which encode some of our highest standards for correct reasoning — the theories of expression and proof which are operative in geometry and technical logic — the representations which are currently sanctioned are uniformly sentential. This tradition is so pervasive that it is rarely explicitly commented upon...the overall sentential style is typically presented as if there were no sensible alternative.[10]

Symbolic logic is purely virtual, a feature that allows classical logic to ignore time, space, energy, causation, context and the rest of reality. Logic and deduction rely solely on fabrications of our imagination. Math historian Israel Kleiner observes that "mathematics evolved for at least three millennia with hardly any symbols."[11]

The **syntax/semantics** barrier of symbolic logic completely separates the meaning of words from the form of the words themselves. The barrier requires that conceptual thought be transcribed into an arbitrary symbolic notation, intentionally leaving the representation of symbolic logic disconnected from our biological evolution and from our natural physiological capabilities. This is perhaps understandable since the textual appearance of words themselves bears no resemblance to the story they tell. However the words of a story can become enriched by their context, their ambiguity and their nuance while the symbols of logic do not tolerate any ambiguity or nuance.

Mark Greaves
c1960–

Israel Kleiner
1937–2024



2.2 Postsymbolism

Visual and interactive media have become dominant modes of communication. Yes I am typing words at this moment, but the context that these words are being born into is no longer a sterile sheet of paper. Their current digital environment is richly infused with images, videos, display windows, editing icons, dynamic streaming channels of non-textual information, verification and citation tools, automated spelling and grammar checkers, summarization tools, and pointers skittering over the pixelated terrain. Long descriptions are being supplemented and even replaced by the *show-me* of photographs and videos. Gregory Bateson, a founder of the field of cybernetics, observes

> ...our iconic communication serves functions totally different from those of language and, indeed, performs functions which verbal language is unsuited to perform.[12]

Iconic

An icon bears a resemblance to the ideas it is intended to convey. The structure of an icon reflects its intention. Iconic logic has a look and feel that connects reasoning to our bodies as well as to our minds. Iconic language then has the additional (many would say non-mathematical)

Gregory Bateson
1904–1980

symbolic logic	☞	*unary logic*
strings of words and symbols		spatial images
linear arrangement of symbols		deletion and creation of icons
required symbolic expressions		existent possibilities
accumulation of facts		elimination of the irrelevant
memorization		recognition
conceptualization		experience
mind/body dichotomy		visceral thinking
goal-directed conclusions		constraint-directed potential
truth-value dualism		the unity of existence

Figure 2-1: *Changes in Perspective*

requirement of supporting both sensuous and concrete understanding. Figure 2-1 summarizes some of the changes in perspective introduced by unary logic.

Gottfried Leibniz
1646–1716

Leibniz studied and strongly advocated the use of images and icons. "The best signs are images; and words, insofar as they are adequate, should represent images accurately."[13] Other examples of iconic logic include Aristotle's Square of Opposition, Venn diagrams, Karnaugh maps, visual programming languages, concept maps and semiconductor circuit schematics. The mathematical systems of lattices, cellular automata, knot theory, topology and category theory all incorporate icons.

Icons stand independently from one another. They are more like paragraphs and stories than nouns and verbs. Once we get comfortable with the idea that structural transformation of icons can clarify their intention, our eyes and fingers can guide reasoning. Symbols can be rearranged but icons can be *animated*.

Embodied

Unary logic is visceral. It can be seen and touched and experienced. The *embodiment* of logic empowers new models of thinking and new methods for the attainment of knowledge. Stanford mathematics professor Keith Devlin asserts that formal concepts cannot be abstract.

Keith Devlin 1947–

> The brain can think only in real, concrete terms. We do not learn to think abstractly; we learn to make the abstract appear real. We do not extend into a new realm, we pull back into a familiar one.[14]

Here's science and philosophy professor Timothy Lenoir:

Timothy Lenoir 1948–

> Thought, including abstract thought such as mathematical reasoning, rests on metaphors and diagrams derived from repeated and deeply layered patterns of body movement.[15]

The properties we impose upon abstractions are invariably associated with our *bodies*, putting physical perception at the foundation of iconic representation. Scale, orientation and location fundamentally refer to us. Mathematics educator Paul Lockhart emphasizes that we make distinctions that impose features upon an empty space that are not inherent in the space (or *void*) itself.

Paul Lockhart 1961 –

> Space has no orientation, no natural unit, and no special place. There is no such thing as left or right, up or down, big or small, here or there, until we make these choices.[16]

Francisco Varela, a pioneering biologist, cybernetician and neuroscientist at the University of Paris explains:

Francisco Varela 1946–2001

> The proper units of knowledge are primarily *concrete*, embodied, incorporated, lived.... The concrete is not a step toward something else; it is both where we are and how we get to where we will be.[17]

Scale, orientation and location fundamentally refer to our physical experience. Tristan Needham, in his ground-breaking textbook *Visual Complex Analysis* makes this appeal:

Tristan Needham c1960–

> When one opens a random modern mathematics text on a random subject, one is confronted by abstract symbolic reasoning that is divorced from one's sensory experience of the world....The present

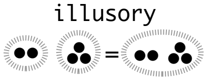
book openly challenges the current dominance of purely symbolic logical reasoning by using new, visually accessible arguments to explain the truths of elementary complex analysis.[18]

SIGNS

symbolic
2 + 3 = 5
memorize

iconic

manipulate

illusory

imagine

Iconic techniques support the postsymbolic perspective that **cognition is embodied**, that formal rigor arises from a biological and physiological context. For some, mathematical thought has always been postsymbolic. Here's Einstein:

> Words and language, whether written or spoken, do not seem to play any part in my thought processes. The psychological entities that serve as building blocks for my thought are certain signs or images, more or less clear, that I can reproduce and recombine at will.[19]

Unity

Western rationality is built upon the grounds of TRUE and FALSE. Duality is at the heart of our world view. In contrast, unary logic incorporates only one ground constant. When truth-values are no longer relevant, *truth becomes relevance*.[20] Unresolved variables stand in place of possibilities. Forms without variables condense to the only foundation ( ) or they cease to exist. While deduction is goal-directed and seeks truth as a result, boundary minimization is constraint-directed and seeks to identify all that is possible. The existence of a mapping between these two fundamentally different expressions of logic indicates that the concepts of logic are purely structural. As emphasized by both Wittgenstein and early Carnap and by postmodernism in general, TRUE and FALSE, as well as the logical connectives, have no meaning other than the webbing of their mutual construction.

Humberto Maturana
1928–2021

The iconic ground arises out of framing nothing. The act of making a distinction brings the frame into existence. *Bounded nothing* provides a single conceptual foundation from which to understand formal logic. Here are cyberneticians Maturana and Varela:

> A unity (entity, object) is brought forth by an act of distinction. Conversely, each time we refer to a unity in our descriptions, we are implying the operation of distinction that defines it and makes it possible.[21]

OBJECTIVE
symbolic objects
`TRUE` `FALSE`
structural processes
←|→

Symbolic logic assumes an *objective perspective* by seeing `TRUE` and `FALSE` as opposing domains, each domain with equivalent possibility. The reasoner is an abstracted observer external to the form of truth. Change, in the form of implication or computation, takes place in a separate temporal dimension. Unary logic takes the *subjective perspective*. The reasoner/participant can see the inside from the outside. Truth and falsity are differentiated only by our frame of reference, by our choice of perspective. Objectively they both arise from the same substrate.

SUBJECTIVE
iconic boundary
( )
structural viewpoint
→(→)

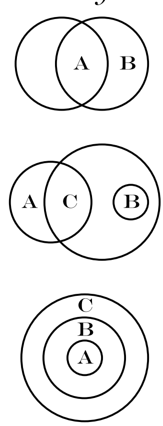
2.3 Pioneers

Boundary logics are not new. They were incipient in the thoughts of both Leibniz and Euler.[22] The first use of closed planar circles to represent syllogistic relations was probably in the 16th century.[23] In the eighteenth century, Euler proposed a diagrammatic foundation for logic based on spatial enclosure. His **Euler diagrams** associate embedded and overlapping circles with the syllogisms.[24] Euler explains:

Leonhard Euler
1707–1783

> The foundation of all these forms is reduced to two principles, respecting the nature of *containing* and *contained*. I. Whatever is in the thing contained must likewise be in the thing containing, and II. Whatever is out of the containing must likewise be out of the contained.[25]

Euler diagrams



Origins

Eleventh century scholarship was defined by the seven liberal arts. The *quadrivium* of arithmetic, geometry, music and astronomy preceded what we now view as the mathematical sciences. The *trivium* of grammar, rhetoric and logic (i.e. the validity of dialectic) preceded what we now view as the proper use of language. Logic was thus inextricably tangled with the linear sequential structure of reading, writing and speaking words. Logic and rationality followed then as part of language rather than mathematics.[26] In the early nineteenth century the discovery of non-Euclidean geometry shook the mathematical world so fundamentally that the ancient Greek perspective of deriving mathematical knowledge from diagrams and physical manipulatives was abandoned in favor of purely symbolic approaches.[27] Mathematics and logic were both extensively revised and redefined to embody **formal structure**. Here's philosopher Brian Rotman:

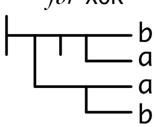
Brian Rotman
1938–2026

> Within the Platonist program, this alphabetic prejudice is given a literal manifestation: linear strings of symbols in the form of normalized sequences of variables and logical connectives drawn from a short, preset list determine the resting place for mathematical language in its purest, most rigorously grounded form.[28]

George Boole
1815–1864

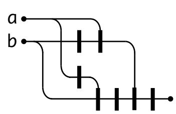
Symbolic logic expressed as an algebra was first introduced by George Boole in 1854. The original explorers of logic developed their understanding of formalization with iconic rather than symbolic forms. Gottlob Frege was the first to formalize logic. In 1879 he published his *Begriffsschrift*, which translates as *concept writing*.[29] Frege developed his own iconic form, **Frege diagrams**. A stylized version of Frege diagrams that maps directly onto parens forms is presented in Chapter 14 as the *crossbar* dialect.

Frege diagram
for XOR



Boundary logic was explicitly defined by Charles Sanders Peirce, published first in 1896 but shared with colleagues in 1882. Peirce considered his boundary logic system of **Existential Graphs** to be his greatest accomplishment.[30]

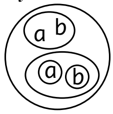
crossbar dialect



> The existential graphs are a system of logical notation which enables the reasoner to judge at a glance of the validity of his processes.[31]

Peirce believed that logic, and ultimately thought, was iconic rather than symbolic, geometric rather than textual.

Peirce graph
for XOR



> All necessary reasoning is diagrammatic. ...all reasoning depends directly or indirectly upon diagrams.[32]

For Peirce rational thought was directly observable in the iconic structure and transformations of boundary logic. Text *obscures* rational thinking by hiding its spatial structure behind one-dimensional sequences of arbitrary tokens. Peirce was an enthusiastic advocate of postsymbolic logic, including the *animation* of boundary forms a century ahead of its time. Boundary logic was *the form of rigorous thought* while its animation would be seeing thought in action. "Thought is the process of animating signs."[33] More generally Peirce believed that "The method of mathematics is the study of diagrams or graphs."[34] In contrast philosopher Neil Tennant wrote:

Neil Tennant
1950–

> [A diagram] is dispensable as a proof-theoretic device...proof is a syntactic object consisting only of sentences arranged in a finite and inspectable array.[35]

Mathematician Philip Davis provides some details:

Philip Davis
1923–2018

> ...for two centuries mathematics has had harsh words to say about visual evidence. The French mathematicians around the time of Lagrange got rid of visual arguments in favor of the purely verbal-logical (analytic) arguments that they thought more secure.[36]

Joseph-Louis Lagrange
1736–1813

Dehumanization

Hilbert's influential program at the turn of the twentieth century set out to express mathematical reasoning solely in finite strings of symbols. Whitehead and Russell's symbolic notation in *Principia Mathematica* was adopted universally after around 1910.[37] During the first half of the twentieth century Hilbert, Carnap and the Vienna school of logical positivists supported symbolic formalization and its more general perspective of structuralism.[38] Their central idea was to detach the notation of mathematics in general and logic in particular from interpretation. The concepts of logic were defined only by the structural relationships between encoded symbols, without regard for any association with reality. Carnap:

Alfred N. Whitehead
1861–1947

Bertrand Russell
1872–1970

David Hilbert
1862–1943

Rudolf Carnap
1891–1970

> A theory, a rule, a definition, or the like is to be called formal when no reference is made in it either to the

> meaning of the symbols (for example, the words) or to the sense of the expressions (e.g. the sentences), but simply and solely to the kinds and orders of the symbols from which the expressions are constructed.[39]

John Venn
1834–1923

Euclid
c. 450–350 BCE

Herbert Simon: "Rigor, it was believed, called for reasoning to be formalized in symbols arranged in sentences and equations."[40] The iconic notations of Euler, Peirce, Frege, Venn and even Euclid were seen at best as only syntactic variants that contributed nothing to understanding, and at worst as sources of error and confusion. Stanford logic professors Jon Barwise and John Etchemendy:

Jon Barwise
1942–

John Etchemendy
1952–

> ...despite the obvious importance of visual images in human cognitive activities, visual representation remains a second-class citizen in both the theory and practice of mathematics.[41]

This position was so extreme that in the 1960s cognitive scientists believed that mental imagery was an epiphenomenon generated by neural processing of symbols. The advent of the digital computer strongly reinforced the linear, bit-stream model of logical computation. Computer science too, by default, is structuralist since silicon computation does not recognize the human concept of meaning. What is asserted as a axiom in symbolic logic is implemented as a string manipulation rule in computer science.

Resurgence

The formalization of mathematics, the acceptance of behavioral psychology and the educational practice of rote memorization each supported the idea that *meaning* was not a rigorous or scientific concept. In the latter half of the twentieth century, these beliefs gradually shifted. Symbols were seen to require augmentation, meaning was to be added separately. And the ban on diagrammatic formal thinking began to weaken. By the 1990s, diagrammatic and textual representations were being compared. Here's Larkin and Simon:

Jill Larkin
1943–

> The fundamental difference between our diagrammatic and sentential representations is that the

> diagrammatic representation preserves exactly the information about the topological and geometric relations among the components of the problem, while the sentential representation does not.[42]

Barwise and Etchemendy have built logic tutorial courses based on *heterogeneous logic,* symbiotically using sentential formulas to reason about diagrammatic scenarios. They argued against any universal scheme of representation:

> Diagrams are physical situations. They must be, since we can see them....the user can simply read off the facts from the diagram as needed. This situation is in stark contrast to sentential inference, where even the most trivial consequence needs to be inferred explicitly.[43]

Their Hyperproof instructional software

> ...is a deductive system that takes seriously the heterogeneity of the information we deal with in everyday life. It allows us to present problems which combine graphical and sentential information, and gives the user a stock of logically valid rules for integrating these different forms of information.[44]

Jaron Lanier is motivated by the potential of *virtual reality,* a more participatory but still virtual interface to concept.

> We'd then have the option of cutting out the 'middleman' of symbols and directly creating shared experience. A fluid kind of concreteness might turn out to be more expressive than abstraction.[45]

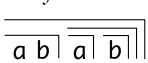
Void-based Form

The next major contribution to boundary logic was Spencer-Brown's *Laws of Form,* a mathematical text that caused considerable excitement by introducing void-based logic within an algebraic framework.[46] Spencer-Brown updated Peirce's Existential Graphs while also enriching the philosophical and metaphysical perspectives of thinking with boundaries. He sought to elucidate the foundational canons of mathematics, building from the

concept	***contrast***
existence	Every concept must have a symbolic representation. — *Some representations can be beneficially tossed into oblivion.*
symbols	We cannot trust diagrams and non-symbolic structure. — *We can trust postsymbolic communication.*
linearity	Symbols must flow sequentially, in a linear string. — *Forms can be accessed in parallel, all at the same time.*
arrangement	Information is transacted by rearrangement of symbols. — *Containment is sufficient for structure.*
location	The relative location of symbols carries their meaning. — *Some forms have no location and some have no meaning.*
memory	Deletion is loss of information. Forgetting is undesirable. — *Deletion of irrelevance improves comprehension and rationality.*
duality	Communication requires the contrast of two symbols. — *A single icon is sufficient for communication and understanding.*
knowledge	Knowledge is the accumulation of symbolic structure. — *Knowledge grows from the elimination of irrelevancy.*

Figure 2-2: *Heresies of Unary Logic*

Spencer-Brown
Laws of Form XOR



first distinction drawn within *void*.[47] Influential mathematicians who reviewed *Laws of Form* have concluded that Spencer-Brown's algebra of distinctions is isomorphic (structurally identical) to Boolean algebra. Most academics have failed to appreciate the fundamental nature of the concepts introduced by Peirce and by Spencer-Brown, demoting the iconic innovations to the role of a notational variant of a formal system they are familiar with.[48] But the changes are deeper than notation. They impact what representation means, what can validly be called notation, which structures legitimately belong to logic and which have been imposed by mathematical formalization, what supports the unity of logic and what supports logical dualism, how knowledge can be acquired, and how rationality and clear thinking might work. As emphasized by Whitehead in 1936, "Fundamental progress has to do with the reinterpretation of basic ideas."[49]

To achieve a semblance of simplicity, unary logic violates many of the conventional rules that define formal notation. Figure 2-2 briefly summarizes some of these revolutionary changes. Logic without the dualistic grounding of `TRUE/FALSE` and without the disembodied abstraction of symbolic words is still, as it turns out, logical.

2.4 Representation

Notation, how we record structural ideas, is both a problem and a solution. Notation directly influences the formulation and understanding of formal concepts. I'll use the word *notation* to refer to symbolic representation and the word *dialect* to refer to iconic representation.

Textual Delimiters

The parens textual delimiter ( ), has no relationship to any geometric shape that its appearance may invoke. *Parens represent generic frames*. Different occurrences of quasi-iconic parens containers, such as the four occurrences of ( ) within the form (( )(( ))), can be distinguished by indices, such as $((\ )_2((\ )_4)_3)_1$, in order to more easily express containment relations in text. Parens delimiter 1 frames subforms 2 and 3.

$((\ )_2((\ )_4)_3)_1$

1 contains 2
1 contains 3
3 contains 4

Pattern-variables

Capital letters such as A, B and C are pattern-variables that represent any configuration of forms regardless of complexity. Pattern-variables may possibly be void-equivalent, identifying no forms at all. The unary axioms Occlusion and Involution do not require matching of pattern-variables. These axioms address *transformation of specific boundary structures* rather than forms in general. Pervasion is the only axiom that requires structural pattern-matching of arbitrary forms. Chapters 3 and 5 include more extensive descriptions.

occlusion
(A ( )) =

involution
((A)) = A

pervasion
(A ⁝ A ⁝) = (A ⁝ ⁝)

Symbolic languages are designed specifically for *breadth* of notation. They do not accommodate *depth* of nesting.[50] Symbolic transformation rules assume that nesting depth is intentionally specific, managed by functions and relations with specific scope and arity. There is no common symbolic notation for an arbitrary depth of nesting, for an arbitrary number of function iterations, or for the paths threaded through formulas by variables. The **vertical ellipses ⁝ ⁝** in Pervasion indicates *depth* independence.[51] This porous vertical boundary is an icon for the transparency of all logic boundaries from the outside. It has the dual utility of implicitly representing non-participating structure at any depth while also identifying replicas of the host form A that are void-equivalent.

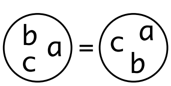
Incidental Structure

One potential confusion in mapping between iconic and symbolic languages is the attribution of *incidental* (extraneous, irrelevant, inadvertent) structure to iconic forms.[52] For example, *commutativity* is one of the fundamental ideas in the symbolic notation of group theory. Commutativity of the connective AND would be expressed as

P AND Q = Q AND P

The introduction of sequence within a conjunction is an *incidental artifact* of recording the conjunction of propositions as a string of symbols. *Logically* AND should not be required to make the distinction of an ordering.[53]

In unary logic any assumed sequence of forms within a space is an incidental imposition of structure from a different formal system. Apparent linear variation of structure can be ignored as an incidental artifact by *reading* rather than by rule. For example, the equation



(a b) = (b a)

includes a potentially misleading use of the sign of equivalence: variation in typographic structure is incidentally confused with conceptual intention.[54] (a b) and (b a) are *identical*. Unary logic does not include *implicit* commutativity. It is a system within which the idea of commutativity is not only irrelevant, it is non-existent. Francisco Varela and Louis Kauffman, an emeritus

Louis Kauffman
1945–

professor of mathematics at the University of Illinois, Chicago, have independently and jointly contributed fundamental innovations to the *Laws of Form* perspective. Both have deeply influenced the development of unary logic. Here is their perspective:

> [Containment] has no particular order properties.... it is not so much that commutativity and associativity are tacitly assumed, but that we are articulating a level at which they do not yet exist![55]

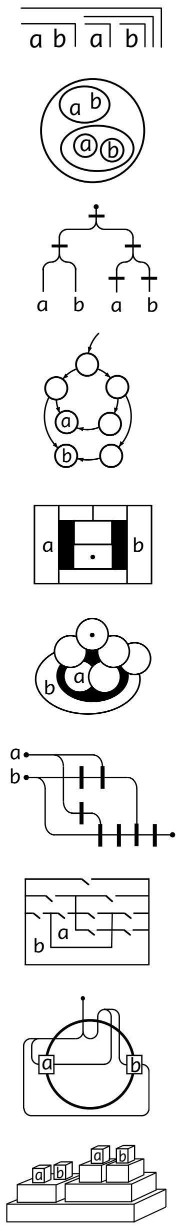
Normally this amount of care in the use of notation is unnecessary, however in shifting perspectives between boundary forms and textual formulas it becomes crucial. Parens delimiters, for example, introduce considerable incidental structure of their own. An iconic representation of boundaries does not support fragmentation of containers into two (left-open and right-close) delimiters. The location of the edge of a boundary is not constrained to one or two dimensions.

There is a more abstract perspective. The representational space within which icons are recorded is topological rather metric. Topology supports containment but not shape, distance, direction or ordering.

Metalanguage

To explore the relationship between images and words we will need a *metalanguage*. A **metalanguage** is a language that stands outside of a formal system, to introduce explanatory descriptions, analogies, metaphors and conceptual connections that are not *within* the system being described. A metalanguage is necessary to describe other languages.[56] The description of any void-equivalent form requires metalanguage. While reading, for example, are we to interpret a word as its meaning or as its representation? An apple is a fruit; the word “apple” has five letters. The apple that “apple” refers to may be red, but “apple” is read. In the case of unary logic, does a parenthesis indicate a parenthetical comment or does it stand in place of a formal container boundary?

((a b)((a)(b)))



I'll use different fonts to keep track of words and tokens that belong to different languages. Our conversational metalanguage, to which these current words belong, is presented in Cochin typeface. Forms within the formal language itself is rendered in Monaco font. ( ) is Cochin while `( )` is `Monaco`.

The Finger

We will often transcribe between two different representational (and conceptual) systems, usually between the conventional strings of logic formulas and the iconic forms of unary logic. I'll introduce a mapping symbol, ☞, to indicate when we are changing systems.

one formal system ☞ *another formal system*

We'll restrict the use of ☞ to cases for which there is a known **transcription map** between the two systems. This makes ☞ a type of equivalence sign (a morphism). We'll call it *the finger*. The pointing finger is also a reminder that a *cognitive shift* is necessary. Transcription is broader than mapping the same concept across two different notations. The finger makes us aware that transcription from symbolic formulas to unary forms requires a reorientation to the form and meaning of both representation and logic.

Independence

In unary logic, every form is *singular*. There is only a single object at the outermost level of a complex form (or possibly the outermost level may be empty). Should more than one form be left as residual at the outermost level a void-equivalent nested-pair is always available to make the single outermost frame explicit. There is another unusual notation that may cause an incidental expectation. When void-equivalence is being asserted, as in the Occlusion axiom, the right-side of the equality will be empty. I have inserted the token *void* in place of emptiness for reading comfort: `(A ( ))` = *void*. The token is from the metalanguage and is not a part of any boundary expression. Please ignore its apparent existence.

dialect	*features*
Enclosures	Visualization of nesting
Trees	Paths composed of nodes and links
Distinction networks	Parallel processing, structure sharing
Maps	Traversal by border crossing
Stepping stones	3D projection of depth as stairs
Crossbars	Flow through barriers
Rooms	Environmental exploration
Paths	Experiential participation
Blocks	Physical manipulation

Figure 2-3: *Participatory Features of Iconic Dialects*

2.5 Unary Dialects

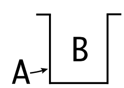
Boundary languages are multidimensional. We will see several in Chapter 14. *Distinction networks* (**dnets**) are the most prevalent. Although boundary forms are conveniently represented by two dimensional images on the page, boundary languages include a variety of modalities. It is not the concepts represented by unary logic that are multidimensional, it is representation itself that has different spatial dialects for the same concepts, analogous to different spoken languages and to different shapes for a generic triangle. *Containment relations* can also be expressed as maps, networks and physical relationships such as touching, stacking and traversing. Figure 2-3 lists some of the participatory features of the dialects introduced in Chapter 14 and displayed in the opposing margin. From the perspective of symbolic logic, each dialect is an **application** or a **model** of the symbolic encoding. From an iconic perspective semantics and representation are merged. The potential physicality of containers means that we can animate, manipulate, interact with, and inhabit boundary forms. There is no abstract/concrete dichotomy. There is no mind/body split. There is no structural dualism.

MULTIMODAL

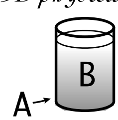
1D symbolic

A contains B

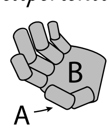
2D iconic



3D physical

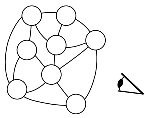


4D experiential

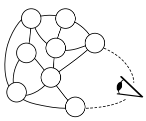


2.6 Remarks

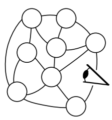
objective



subjective



participatory



Symbolic logic is obscure not by its own fault but by its historical embrace of the narrow perspective of viewing structure only from the outside. This can be understood as a still remaining residual of the Ptolemaic exceptionalism (c.1500) that we live at the center of the universe. Linear perspective was enshrined by the Gutenberg revolution (1440) that propagated text rather than images. The outside perspective, also called objectivity and also called the God's eye view, assumes prescience. The inside perspective acknowledges that a structure is constrained by its environment, that the reader is constrained by the same environment, and that understanding requires epistemological humility.

How we think is entirely personal. A unary approach has the potential to recognize formal systems from a more humane perspective. Computation is a feeling as well as a symbolic consequence. Written fiction can transport us into different worlds and different emotions. Music has an even deeper grounding in emotional response. We *can* feel validity, but formal symbols constrained within a straightjacket of mathematical certainty are very poor conductors of emotion. Unary logic has the potential to allow us to escape truth-values as the conceptual arbiter of physical reality.

Laws of Form has launched us into a postsymbolic mathematics where spatial forms condense symbolic complexity, where the syntax/semantics barrier dissolves, where objects are united with processes, where absence is a primary conceptual tool, where representation is both visceral and sensuous, and where the viewing perspective of the reader is directly implicated within the form. When applied to logic, Spencer-Brown's boundary logic challenges a foundational assumption of Western thought, that rationality requires dualism. In the following chapters we explore the utility of void-equivalence for deduction and computation, and illustrate the use of semipermeability and parallelism as features of logic boundaries.

Endnotes

1. **opening quote:** Spencer-Brown p.xx.

2. **ability to formally manipulate symbols:** A. Newell & H. Simon (1972) *Human Problem Solving*. This book lead to the *information processing model* of reasoning that dominated cognitive science for several decades. During the 1990s algorithmic computer reasoning was considered to be a model of human reasoning. "Today...there is a debate as to whether imagery, *as a distinct form of representation*, plays *any* role in cognition." R. Lindsay (1995) Images and Inference. In B. Chandrasekaran, J. Glascow. N. Hari Narayanan (eds) *Diagrammatic Reasoning* p.111.

3. **and not an inevitability:** The reference here is to Western European mathematical tradition. The field of ethnomathematics has documented many cultural uses of mathematics that do not align with symbolic dualism.

4. **wires during silicon computation:** 100 billion transistors (10^{11}) clocked at 3 gigahertz (3×10^9) is over 10^{20} transistor activations per second.

5. **the *structure* of symbolic deduction:** Although unary logic does pose significant philosophical questions and alternatives, I am strictly avoiding philosophical debate by focusing on the *structure of representation*, without interpretation and without attempting to determine *why* iconic form is functionally equivalent to symbolic form.

6. ***postsymbolic communication*:** Jaron Lanier (2010) *You are Not a Gadget*.

7. **it must stand for *something***: Julian Barbour (2011) Bit from it. (italics in original) Online 12/25 `http://platonia.com/bit_from_it.pdf`

8. **experience within a physical environment:** It is scandalous, in my opinion, that our educational institutions have marginalized these universal features of human existence by limiting teaching to only symbolic skills.

9. ***textual* forms of the word "triangle":** The 36 textual languages for the word "triangle" in the margin are:

1. Latin, 2. Hawaiian, 3. Chinese/Japanese, 4. Malay, 5. Dutch, 6. Nepali/Hindi, 7. Russian, 8. Georgian, 9. Danish, 10. Farsi, 11. Xhosa/Zulu, 12. French, 13. Hebrew, 14. Hungarian, 15. Greek, 16. Spanish, 17. Telugu, 18. Ukrainian, 19. Norwegian/Swedish, 20. Mongolian, 21. Amharic, 22. German, 23. Bengali, 24. Welsh, 25. Malayalam, 26. Icelandic, 27. Korean, 28. Somali, 29. Kannada, 30. Tatar, 31. Polish, 32. Khmer, 33. Albanian, 34. Uyghur, 35. Turkish, 36. Myanma.

10. **as if there were no sensible alternative:** M. Greaves (2002) *The Philosophical Status of Diagrams* p.2.

11. **millennia with hardly any symbols:** I. Kleiner (1991) Rigor and proof in mathematics: A historical perspective. *Mathematics Magazine* 64 p. 291-314. Online 1/26 `https://www.tandfonline.com/doi/abs/10.1080/0025570X.1991.11977625`

12. **is unsuited to perform:** G. Bateson (1972) Redundancy and coding, in *Steps to an Ecology of Mind* p.411ff. Online 8/18 `http://shifter-magazine.com/wp-content/uploads/2015/11/gregory-bateson-steps-to-an-ecology-of-mind-1.pdf`

13. **represent images accurately:** Leibniz to Tschirnhaus, 1679 (Math., IV, 481; Brief., I, 405), as quoted in L. Couturat (1904) *The Logic of Leibniz*, Ch. 4 footnote 93.

14. **pull back into a familiar one:** K. Devlin (2006) The useful and reliable illusion of reality in mathematics. *Toward a New Epistemology of Mathematics Workshop*, GAP6 Conference 9/2006. Online 8/18 `https://web.stanford.edu/~k-devlin/Papers/Berlin06.pdf`

15. **layered patterns of body movement:** T. Lenoir (2008) Machinic bodies, ghosts, and paraselves: confronting the singularity with Brian Rotman. Foreword to B. Rotman (2008) *Becoming Beside Ourselves* p.xxv.

16. **until we make these choices:** P. Lockhart (2012) *Measurement* p.206.

17. **how we get to where we will be:** F. Varela (1992) *Ethical Know-How: Action, Wisdom, and Cognition* p.7. (Emphasis in original.) Online 8/18 `https://www.heartoftheart.org/wp-content/uploads/2017/08/Varela-F.-J.-1999-Ethical-know-how.-Action-wisdom-and-cognition-2119.pdf`

18. **complex analysis:** T. Needham (1999) *Visual Complex Analysis* p.vii.

19. **reproduce and recombine at will:** Einstein, quoted by K. Devlin (2006) *Toward a New Epistemology of Mathematics Workshop*.

20. ***truth becomes relevance*:** The nature of truth is widely discussed in philosophy. For an applied summary of theories of truth consider:

J. Mingers & C. Standing (2018) Framework for validating information systems research based on a pluralistic account of truth and correctness. *Journal of the Association for Information Systems* **19**(4) p.247-265. Online 12/25 `https://arxiv.org/abs/1701.04676`

21. **defines it and makes it possible:** H. Maturana & F. Varela (1987) *The Tree of Knowledge: The biological roots of human understanding.* In a series of books Varela and Maturana apply distinction-based thinking to biological systems. A new model of computation based on biological cell membranes

uses the same containment data structures and similar parallel processing techniques as unary logic. G. Păun (2006) Introduction to membrane computing. In G. Ciobanu, G.Păun & M. Pérez-Jiménez (eds) *Applications of Membrane Computing* (2006) p.1-42. Also see L. Kauffman (2002) Biologic. arXiv:quant-ph/0204007v2

22. **both Leibniz and Euler:** Diagrammatic systems for syllogistic logic appear in *De formæ logicæ comprobatione per linearum ductus* (Phil., VII, B, IV, 1-10) Couturat's edition (Leibniz & Couturat, 1903 p.292-328).

23. **probably in the 16th century:** R. Llull (1501) *Ars magna. Barch[ino]ne: Impressum per Petru[m] Posa*.

24. **overlapping circles with the syllogisms:** From the ancient Greeks to the eighteenth century formalized deduction was exclusively *syllogistic reasoning*, a tightly constrained version of implication with quantification.

25. **must likewise be out of the contained:** L. Euler (1802) Letter CIV Different forms of syllogisms (2/21/1761). *Letters of Euler*. H. Hunter (trans) p.406. Euler's definition of containment can be compared to Spencer-Brown's *Laws of Form* p.7 : "...a cross is said to contain what is on its inside and not to contain what is not on its inside."

26, **rather than mathematics:** I teach a course in symbolic logic at my college. I'm a member of the Mathematics faculty but I have to teach the course out of the Philosophy department. Logic cannot be mixed with mathematics. To my great pleasure my college does not have a Philosophy department, I teach the only course offered by the non-existent department.

27. **purely symbolic approaches:** The word calculus, from Latin, translates to *pebble*, since calculation was done for millennia using manipulative objects.

28. **rigorously grounded form:** B. Rotman (2000) *Mathematics as Sign* p.55.

29. **which translates as *concept writing*:** Online 3/24 https://dec59.ruk.cuni.cz/~kolmanv/Begriffsschrift.pdf

30. **his greatest accomplishment:** S.J. Shin (2022) Peirce's Deductive Logic, Section 2, *Stanford Encyclopedia of Philosophy*. (Summer 2022 Edition). E. N. Zalta (ed) https://plato.stanford.edu/archives/sum2022/entries/peirce-logic. Peirce did not academically document Existential Graphs until 1909 in Peirce MS 514.

31. **the validity of his processes:** Peirce (1909) MS 514, §4.551.

32. **or indirectly upon diagrams:** C.S. Peirce (1904) The new elements of mathematics. In C. Eisele (ed) (1976) *Mathematical Philosophy* 4 p.314.

33. **process of animating signs:** Peirce (1868) § 2.228.

"...the emergence of the 'moving-pictures' idea coincides with Peirce's living in New York 1895-97, where the first commercial films were premiered." A-V Pietarinen (2020) Getting closer to iconic logic. *Diagrammatic Representation and Inference. Diagrams 2020.* Lecture Notes in Computer Science, vol 12169. p.57-58.

34. **the study of diagrams or graphs:** Peirce Volume 3 § 556. Peirce was prescient here since graphs are now the organizational substrate of both the internet and AI neural networks.

35. **in a finite and inspectable array:** N. Tennant (1984) The withering away of formal semantics. *Mind and Language V.* **1**(4) 1986 p.302-318.

36. **they thought more secure:** P. Davis (1997) Mathematics in an age of illiteracy. *SIAM News* **30**(9) 11/97 p.30.

37. **adopted universally after around 1910:** A. N. Whitehead & B. Russell (1910/1980) *Principia Mathematica.*

38. **general perspective of structuralism:** Although all mathematical systems are formal, structuralism was initially developed within literature, sociology and the arts as the study of how meaning might grow out of the rules, abstractions, patterns and structural properties of signs and symbols.

39. **which the expressions are constructed:** R. Carnap (1937) *The Logical Syntax of Language* p.1.

40. **in sentences and equations:** H. Simon, forward to B. Chandrasekaran, J. Glasgow & N. Narayanan, (eds) (1995) *Diagrammatic Reasoning* p.xi.

41. **theory and practice of mathematics:** J. Barwise & J. Etchemendy (1996) Visual information and valid reasoning. In G. Allwein & J. Barwise (eds) *Logical Reasoning with Diagrams* p.3.

42. **the sentential representation does not:** J. Larkin & H. Simon (1987) Why a diagram is (sometimes) worth ten thousand words. In Chandrasekaran et al *Diagrammatic Reasoning* p.70.

43. **needs to be inferred explicitly:** Barwise & Etchemendy. In Allwein & Barwise *Logical Reasoning with Diagrams* p.23.

44. **these different forms of information:** J. Barwise & J. Etchemendy (1994) *Hyperproof.* CLSI Publications p.xiii. Online 1/26 `https://archive.org/details/hyperproof0000barw`

45. **more expressive than abstraction:** Lanier, p.190.

46. **within an algebraic framework:** Spencer-Brown's book has inspired many descriptions, explanations, essays and applications. Only a few research scholars, mentioned throughout this volume, have extended and applied *Laws of Form*. Many more have described their understanding of aspects of Spencer-Brown's innovations in logic, representation, and more generally philosophy. Nearly three decades ago Jack Engstrom listed some of these research areas:

— *mathematics:* topology, self-reference, automata
— *logic:* three-valued, imaginary, paradox
— *numerics:* integer, real, imaginary
— *biology:* autopoiesis, cognitive systems, cybernetics, neural processes
— *electronics:* computation, programming, silicon design
— *sociology:* communication, semiotics
— *philosophy:* transcendence, metaphysics

J. Engstrom (1999) G. Spencer-Brown's *Laws of Form* as a revolutionary, unifying notation. *Semiotica* **125**(1/3)p.33-45.

47. **distinction drawn within *void*:** When asked about the practical applications of *Laws of Form* in an online discussion, Ernst von Glaserfeld replied: "Spencer-Brown gives a lot of insightful advice in his footnotes. One has to read them, because they are very compressed and cannot be further summarized."

48. **system they are familiar with:** W.E. Gould's review in the *Journal of Symbolic Logic* **42**:2 (6/1977) p.317-318 characterizes Spencer-Brown's work as "...simply another axiomatization of Boolean algebra".

B. Banaschewski in On G. Spencer-Brown's Laws of Form, *Notre Dame Journal of Formal Logic* 18:3 (7/1977) p.507-509, says that "*primary algebra is exactly the theory of join and addition (= symmetric difference) of Boolean algebras*"(italics in the original).

D.G. Schwartz (1981) Isomorphisms of Spencer-Brown's Laws of Form and Varela's Calculus of Self-Reference, *International Journal of General Systems* **6**:p.239-255, establishes "Exact isomorphisms with formal systems which employ the standard linguistic notations".

49. **reinterpretation of basic ideas:** A. N. Whitehead (1936) Harvard: The Future. *The Atlantic Monthly* Sept 1936.

50. **accommodate *depth* of nesting:** There is a notation for nested functions, such as `f(x, f(y, g(z)))`, however this notation refers to a specific structure with a specific scope and not to a generic pattern of nesting.

51. **indicates *depth* independence:** The vertical ellipsis is available in some fonts as the **tricolon**, which is a Unicode general punctuation mark. The Monaco font used for boundary forms in this volume includes the tricolon mark:

`U+205D` Monaco ⁝

The vertical ellipsis is also included as a Unicode mathematical operator symbol. The math symbol is included in fonts such as STIX two math:

`U+22EE` STIX two math ⋮

Finally, three vertical dots is within the Braille 6-dot font in two ways, as dots 123 and as dots 456:

`U+2807` Braille ASCII "L" ⠇
`U+2838` Braille ASCII "_" ⠸

52. **structure to iconic forms:** Shin introduces the concept of *accidental properties* in S.J. Shin (2002) *The Iconic Logic of Peirce's Graphs* p.29. However she introduces accidents as properties that an iconic representation might have that do not represent facts within its interpretation. "These are accidental properties that the adopted icon has but that the denotation of this icon does not." It is dubious that a denotation itself is completely free of its own accidental symbolic properties, we have learned to be blind to them due to the ubiquitous symbolic standard. I am using the idea of *incidental* structure in the opposite direction, as properties that the icon *does not have* but that are attributed to it in an attempt to align it with properties of its symbolic interpretation.

53. **the distinction of an ordering:** The well-known logician W.V. Quine has a similar attitude: "...conjunction...is *commutative*; i.e., order is immaterial, there being no need to distinguish between '*pq*' and '*qp*'. W.V. Quine (1982) *Methods of Logic*. Fourth Edition p.10. (italics in original)

54. **with conceptual intention:** Sequence within unary forms is represented by *depth* of nesting: `(a (b))` ≠ `(b (a))`. Similar to set theory, a collection of forms contained by the same outer frame, as in `(a b)`, is not a sequence.

55. **at which they do not yet exist:** L. Kauffman & F. Varela (1980) Form dynamics p.182. *J. Social Biol. Struct.* **3** p.171-206.

56. **necessary to describe other languages:** Metalanguage is at the center of modern mathematics, since metamathematics attempts to fully describe mathematics itself within the language of mathematics. There are several issues. Logical connectives are so embedded into our understanding of words that they are used to define themselves. For example, the truth of the expression `A AND B` depends upon the truth of `A` *and* the truth of `B`.

Chapter 7

Tautology

CHAPTER CONTENTS

Tautology and contradiction are not pictures of the reality. They present no possible state of affairs.[1]
— Ludwig Wittgenstein (1922)

Ludwig Wittgenstein
1889–1951

The truth is that the foundations of symbolic logic rest upon belief (as does this sentence). Axioms are structures that we wish to believe, that we take as **given**. There is a second source of belief, the **rules of transformation**. Changing the structure of what we take as given yields *deductive knowledge*. Symbolic logic maintains truth-values and generates new knowledge by proof, or at least by convincing demonstration.

This chapter looks at **tautologies**, patterns that can be demonstrated to be TRUE with no supporting facts. Wittgenstein's opening observation is that knowledge requires variation, truth alone is insufficient. *Tautologies are scaffolding*, frameworks that maintain truth independently of any variables they may incorporate. Tautologies hold for all models, for all possible evaluations. Variables within tautologies are place-holders without meaning. These **pseudo-variables** do not support value or variation. Many tautologies are taken to be methods of deduction, others are transformation rules, others are axioms and theorems, some are structural constraints. A central question is: why do we believe tautologies differ from one another?

Chapter 8 examines *axiom systems*, collections of defined tautologies that together construct the base upon which formal logic stands. There are many axiom systems that differ stylistically in how they apply tautological reasoning to deductive demonstration, otherwise they are all quite similar except for technical nuance and historical enculturation. In this chapter I'll employ the unary logic axioms to reduce dozens of symbolic tautologies to the ground, thereby validating them. The complexity of the unary logic reductions provides a *baseline* from which to judge the deductive complexity of the symbolic formulas. The endeavor is quite unique. Propositional logic is taken to be part of the axiomatic basis for almost all mathematical systems. It is itself the ultimate standard of measurement, the custodian of the value of truth.

The unary axioms provide a measure of the relative complexity of tautologies that is external to symbolic mathematics. The goal is not to compare the efficiency of a conventional proof with the efficiency of unary techniques. A glance at any introductory logic book would make it clear that there is no contest. Instead, the goal is to understand conventional tautologies within a new language and a new conceptual structure.

7.1 Unary Transformation

The unary axioms are sufficient to verify the structural validity of all of the tautologies of classical logic. Here first is a summary of their *computational* behavior.

Literals

Atomic variables, represented by small letters like `a`, stand in place of elementary objects. Atomic variables and bounded atomic variables, `(a)`, together are called *literals*. A bounded atomic variable is not more complex than one that is unbounded.[2]

Occlusion as Termination

	`(A ( )) = ` *void*	*create* ⇄ *occlude*	OCCLUSION
☞	`NOT(FALSE IMPLIES A) IFF FALSE`		*logic*
	`NOR(A, NOR( )) ≡ NOR(NOR( ))`		*variadic nor*
	$\neg(\bot \to A) \leftrightarrow \bot$		*symbolic logic*
	$(A + 1)' = 0$		*boolean algebra*

Occlusion is usually the final reduction that ends a computation. Pervasion first empties a container to create a mark and then Occlusion deletes the container of the mark together with all of the contents of that container. More than one application of `occlude` indicates embedded complexity. Reversing the computation, Occlusion is also the beginning, how to bring essentially meaningless and void-equivalent forms into apparent existence.

Involution as Garbage Collection

	`((A)) = A`	*enfold* ⇄ *clarify*	INVOLUTION
☞	`NOT(NOT A) IFF A`		*logic*
	`NOR(NOR(A)) ≡ A`		*variadic nor*
	$\neg\neg A \leftrightarrow A$		*symbolic logic*
	$(A')' = A$		*boolean algebra*

Involution reduces nesting depth by eliminating self-canceling nested-pairs. `Clarify` occurs most frequently during the initial transcription of logic into a unary form. Clarifying nested-pairs by deletion is best done immediately. Computer scientist Michael Beeson:

> Theorems...about double-negation elimination have an intrinsic, aesthetic appeal in that they show the possibility of simplifying proofs, but they also are of interest because they justify in the vast majority of cases a shortcut in automated proof-search methods, namely, the automatic discarding of double negations.[3]

`Clarify` removes the duality that is imported from conventional formulas during transcription. `Enfold` packages multiple forms into a single form, occasionally facilitating more efficient Pervasion.

Pervasion as Deductive Computation

PERVASION

```
(A ⋮A⋮) = (A ⋮  ⋮)      reify ⇄ pervade
 ~~~~~~~~~~~~~
```

logic ☞ NOT(A OR B) IMPLIES A) IFF (NOT (B IMPLIES A)

variadic nor NOR(A, NOR(A, B))) ≡ NOR(A, NOR(B))

symbolic logic $\neg((A \vee B) \rightarrow A) \leftrightarrow \neg(B \rightarrow A)$

boolean algebra $(A + (A + B)')' = (A + B')'$

Pervasion is the deductive engine of unary logic. It is the only axiom that addresses variables. The semipermeable vertical ellipsis eliminates the concept of operator scope and the ownership of particular arguments by connectives and relations.[4] Pervasion literally reaches within a container to delete replicas and to create replicas wherever they might be useful to aid other reductions. In general, existent forms without replicated variables do not reduce further. Pervasion supports reduction while also facilitating the identification of irrelevant and redundant structure.

7.2 Metrics

Figures 7-1, 7-2 and 7-3 show the common tautologies of propositional logic, their unary logic form and the unary demonstration of their validity. Nine metrics are recorded for each tautology. Four are structural descriptions of the unary form of the tautology and five are counts of transformation steps needed to reduce the unary form to mark.

Form:

- ( ): number of containers
- *deep*: maximum depth of nesting
- *var*: number of variable occurrences
- *dvar*: number of distinct variables

Transform:

- *step*: number of parallel reduction steps
- *occ*: number of `occlude` reductions
- *clar*: number of `clarify` reductions
- *per*: number of `pervade` reductions
- *level*: level of complexity for `pervade`

Complexity

Assessing the complexity of a deductive process is necessarily inexact. Any particular deductive problem will have several different techniques available for its resolution, and each technique will support several different sequences of deductive steps. Specific techniques as well are better suited for specific problem types. I'll defer to some of the early metrics used in the design of silicon circuitry.[5] The figures display the structural features of the unary transcription of each tautology and the applications of unary axioms needed to demonstrate the validity of that tautology.

An overarching question is the degree of creativity or insight needed to achieve a demonstration or proof. This concern could be phrased as the ease of developing an algorithmic implementation. With propositional logic however we are on such well-trodden ground that creativity is not a factor. We are instead measuring the *relative structural complexity* of the foundational tautologies of deductive logic, as well as shedding light on their historical use as deductive strategies.

Reduction to Ground

Upon review, raw counts of form and transform appear to offer little understanding. A more refined count better reflects the process of unary reduction:

- One application of `occlude` is necessary to reduce a tautology to ground.[6]
- Applications of `clarify` that can be made in parallel are counted together as one step.
- `Pervade` has several specific levels of effort that generally align with computational complexity.

This refined count also falls short of being informative. Both Occlusion and Involution can be seen as *clean-up* operations that are independent of the presence of variables. Reduction complexity can be associated solely with the levels of effort needed in the application of Pervasion. Four levels can be clearly identified by the number and direction of Pervasion steps needed for reduction:

complexity levels
0: `((A))`
1: `(A (A B (A C)))`
2: `(A (A B)((B) C))`
3: `((A B)(A (B C)))`

— **Level 0:** no application of `pervade` needed
— **Level 1:** immediate reduction by `pervade`
— **Level 2:** direct reduction requiring a sequence of interdependent `pervade` applications
— **Level 3:** reduction requiring initial application of `reify` to trigger later applications of `pervade`

Level 0 tautologies reduce by Involution and/or Occlusion alone. *Level 1* reductions require only one *parallel* application of Pervasion. Pervasion can be seen to be a singular application when multiple replicas of the same subform are concurrently deleted. *Level 2* identifies *multiple* sequential applications of reduction by `pervade`. A single application creates a new structure that then supports a new reduction step. Each application of Pervasion necessarily applies to a different variable. *Level 3* requires the constructive use of `reify` to prepare for Pervasion. `Reify` temporarily increases the complexity of a form, analogous to hill-climbing in order to escape a local minimum. Distribution is the simplest example of a tautology that requires the application of `reify`. Finding a *minimal* demonstration of Distribution (Arrangement) does require some creativity.

arrangement
`A ((B)(C))`
`= ((A B)(A C))`

Overall Results

There are 34 tautologies examined in the following pages. In Chapter 8 there are 22 different axiom *systems*. Both chapters include and reduce about 80 different tautologies. The *level* of Pervasion is most helpful in identifying differential symbolic complexity. The most significant observation is that almost all propositional tautologies, axioms and theorems are *simple* in not requiring Level 3 effort. The following table shows the complexity distribution of more than 100 tautologies that have been considered to be important enough during the last century to be proposed as the foundations of symbolic proof and deduction.

complexity	*tautologies*	*within systems of axioms*	*percent of total*
Level 0	13	9	21%
Level 1	10	33	42%
Level 2	6	19	24%
Level 3	5	8	13%

About 20% of the tautologies are what can be considered from a unary perspective to be trivial symbolic detritus. 13% are technically complex, providing a source of potentially exponential effort. Only two of the tautologies in this chapter reach Level 3 in complexity: Biconditional and Distribution (with 4 different varieties) in Figure 7-3. The other 8 Level 3 axioms in Chapter 8 are successful attempts to create a single axiom basis for propositional logic.

7.3 Tautology Reduction

The details of the analysis are presented in three figures: primary inferential tautologies, primary replacement rules, and other structural rules.

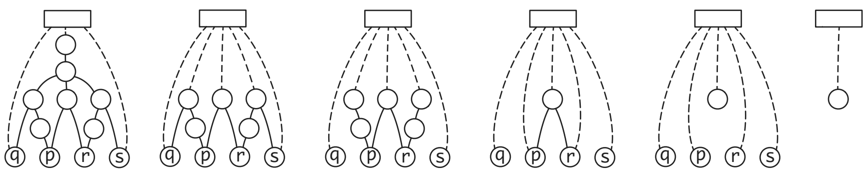
Classical Inference

Figure 7-1 includes eight **inferential tautologies** that are often presented as deductive techniques and strategies.[7] The first three are simple Level 1 applications of `pervade`. This is a surprise given their primary role as deductive techniques in symbolic logic books and courses. The remaining five tautologies have Level 2 complexity, each requires two sequential `pervade` steps.

The following example of Constructive Dilemma illustrates how forms might require a sequence of `pervade` steps. The reduction metrics for this tautology are: *occ*-1 *clar*-1 *per*-2. Quite typically an initial application of `clarify` first deletes the nested-pairs inherited from transcription. Then `q s` is pervaded. Alternatively `pervade` could have been applied first since it is blind to depth of nesting.

```
                          ((((p) q) ((r) s) (p r))) q s
        clarify             ((p) q) ((r) s) (p r)   q s
     pervade q s            ((p)  ) ((r)  ) (p r)   q s
        clarify               p        r    (p r)   q s
    pervade p r               p        r    (   )   q s
        occlude                             (   )
```

Deleting q and s triggers two clarifies that then free p r to also pervade their replicas. This reduction has a metric of *occ*-1 *clar*-3 *per*-2. Here is the same reduction of Constructive Dilemma as a dnet form.



dilemma *clarify* *pervade* q s *clarify* *pervade* p r *occlude*

The formalization of logic took place well before computational parallelism was developed, first with hardware architectures in the late 1950s and later quite widely within the development of multiprocessors in the 1970s. Although Spencer-Brown was obviously experienced with hardware parallelism in 1969, he explicitly defines computational steps in *Laws of Form* to be singular and serial. Spencer-Brown also eschews the linear constraints of symbolic expressions, so his reversion to the linear constraints of transformation is rather anachronistic. Since unary logic defines all forms to be singular and independent, unary transformations are naturally parallel. The *step* metric here is thus misleading in that it measures a symbolic feature of an iconic process.

Classical Equalities

Figure 7-2 includes nine **equational replacement rules**. Each is generally considered to be a structural substitution rather than an inference. From the perspective of unary logic, all nine reductions are close to trivial. The last four have two varieties, one for AND and a second symmetric version for OR. Since the representation of unary

inferential logic rule ☞ ***unary demonstration***

Detachment $(p \wedge (p \rightarrow q)) \rightarrow q$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
5	4	4	2	2	1	1	2	1

```
(((p) ((p) q))) q
  (p) (       )   q
      (       )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Modus Tollens $\neg q \wedge (p \rightarrow q) \rightarrow \neg p$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
7	4	4	2	3	1	1	2	1

```
((((q))((p) q))) (p)
    q   (     q)   (p)
    q   (      )   (p)
        (      )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Disjunctive Syllogism $((p \vee q) \wedge \neg p) \rightarrow q$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
5	4	4	2	3	1	1	2	1

```
(((p q)((p)))) q
  (p  )  p      q
  (   )  p      q
  (   )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Absorption $(p \rightarrow q) \rightarrow (p \rightarrow (p \wedge q))$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
6	2	5	2	3	1	0	2	2

```
((p) q) (p) ((p)(q))
(    q) (p) (   (q))
(    q) (p) (       )
            (       )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Hypothetical Syllogism

$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
7	4	6	3	3	1	1	3	2

```
((((p) q)((q) r))) (p) r
 (     q)((q)  )   (p) r
 (     q)(     )   (p) r
         (     )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Constructive Dilemma

$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee r)) \rightarrow (q \vee s)$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
7	4	8	4	4	1	2	4	2

```
((((p) q)((r) s)(p r))) q s
  ((p)  )((r)  )(p r)   q s
   p       r    (p r)   q s
   p       r    (   )   q s
                (   )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Contradiction $(\neg p \rightarrow (q \wedge \neg q)) \rightarrow p$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
7	4	4	2	3	2	1	1	2

```
(((p)) ((q)((q)))) p
(       ((q)(   ))) p
(                 ) p
(                 )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Cases $((p \vee q) \wedge (q \rightarrow p)) \rightarrow p$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
7	5	5	2	3	1	1	2	2

```
(((p q) ((q) p))) p
(((  q) ((q)  ))) p
  (  q)    q      p
  (   )    q      p
  (   )
```

Figure 7-1: *Unary Demonstration of Inferential Rules*
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

axioms directly transcribes to logical `NOR`, the `OR` versions that express linear orderings require no reduction steps. Every inferential tautology can also be constructed as an equality. This is obvious since the value of a tautology is `TRUE`. Using Detachment as an example:

$$((p \wedge (p \rightarrow q)) \rightarrow q) = \top$$

☞ `(((p) ((p) q))) q = ( )`

Structural Tautologies

Figure 7-3 includes eight **structural rules** that occur often during symbolic proofs. The first five are deductive, the last three are equational. Only Resolution requires sequential Pervasion. The other four deductive formulas, as well as the DeMorgan theorem are simple, Level 0 or Level 1. It may be helpful to compare the initial structures of Constructive Dilemma and Resolution. I've relabeled and rearranged the form of Resolution for a more visual comparison.

constructive dilemma `((((p) q) ((r) s) (p r))) q s`
resolution `((((p) q)         (p r))) q r`

The final two tautologies in Figure 7-3 are Level 3 in complexity. Distribution equates two different but irreducible forms that contain a unate variable. The Biconditional is binate and is the definition of the logical equal sign. It has two different but equivalent irreducible forms. In Chapter 5.3 Distribution is demonstrated as the Arrangement theorem of unary logic and Biconditional is demonstrated as Equality Pivot.

All potential structural complexity resides in these two rules.

Tautologies Summary

The unary axioms alone are sufficient to verify the most common tautologies, many of which are considered to be deductive strategies. The level of complexity is defined solely by Pervasion. With the support of `clarify` Shallow Pervasion is sufficient for all tautologies based on implication and for all equational replacement rules with the exception of Distribution and Biconditional. No tautologies require Absorption.

logic replacement rule ☞ ***unary demonstration***

Excluded Middle $p \lor \neg p = \top$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
1	1	2	1	2	1	0	1	1

```
p (p) = ( )
p ( ) = ( )
  ( ) = ( )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Double Negation $p = \neg\neg p$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
2	2	1	1	1	0	1	0	0

```
p = ((p))
p =   p
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Non-contradiction $\neg(p \land \neg p) = \top$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
5	4	2	1	2	1	0	1	1

```
(((p)((p)))) = ( )
 (p)(     )  = ( )
    (     )  = ( )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Material Implication $p \rightarrow q = \neg p \lor q$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
1	1	2	2	0	0	0	0	0

```
(p) q = (p) q
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Contraposition $p \rightarrow q = \neg q \rightarrow \neg p$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
3	2	2	2	1	0	1	0	0

```
(p) q = ((q)) (p)
(p) q =   q   (p)
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Idempotency $p \lor p = p$ $\quad p \land p = p$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
0/3	0/2	2/2	1/1	1/2	0/0	0/1	1/1	1/1

```
p p = p
  p = p

((p)(p)) = p
((p)   ) = p
  p      = p
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Commutation $p \lor q = q \lor p$ $\quad p \land q = q \land p$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
0/2	0/2	2/2	2/2	0/0	0/0	0/0	0/0	0/0

```
p q = q p

((p)(q)) = ((q)(p))
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Association $p \lor (q \lor r) = (p \lor q) \lor r$
$p \land (q \land r) = (p \land q) \land r$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
0/6	0/4	3/3	3/3	0/1	0/0	0/1	0/0	0/0

```
p q r = p q r

((p)(((q)(r)))) = ((((p)(q)))(r))
((p)  (q)(r)  ) = (  (p)(q)  (r))
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Domination $(p \lor \top) = \top$ $\quad (p \land \bot) = \bot$

( )	*deep*	*var*	*dvar*	*step*	*occ*	*clar*	*per*	***level***
1/3	1/2	1/1	1/1	1/1	1/1	0/0	0/0	0/0

```
p ( ) = ( )
  ( ) = ( )

((p)( )) = void
    void = void
```

Figure 7-2: *Unary Demonstration of Replacement Rules*

All tautologies are *simple* and follow the same simple structural pattern. There is one exception: the Law of Contradiction embeds a contradiction, $q \wedge \neg q$, within its interior, requiring two applications of `occlude`. Five tautologies require a Level 2 sequence of two `pervade` operations (Absorption, Hypothetical Syllogism, Constructive Dilemma, Cases and Resolution).

The descriptive metrics are not particularly useful, other than a general observation that the Level 2 tautologies incorporate more variables. In all cases but Addition, the number of Pervasion steps is the same as the number of different variables.

Distribution (Arrangement) and Biconditional (Equality Pivot) both introduce Level 3 computational complexity. Both of these equational replacement rules can be usefully added to the unary reductive tools as convenience theorems. The complexity of Arrangement and Pivot is necessary, of course, to rescue propositional logic from triviality.

THE TOOLKIT
occlusion
involution
pervasion
absorption
arrangement
pivot

Symbolic logic is taught, particularly in textbooks, as a collection of deductive strategies. Direct proof by natural deduction includes around eight inferential rules/strategies and around ten equational replacement rules, all of which are included in this chapter. Specialized techniques include Conditional Proof and Indirect Proof. From the perspective of unary logic, the strategies and techniques of logical proof are *not different*. None provide specialized advantages. The entire collection of about twenty apparently different symbolic logic tools can be reduced to three deletion/creation axioms, one deletion/reduction theorem, and a couple of rearrangement theorems.[8]

7.4 The Constructive Perspective

Reversing the steps of a demonstration constructs each tautology from the ground. This perspective highlights how almost all logical tautologies have the same structural origin. Contradiction is the exception. Detachment provides a generic stepwise template for constructing tautologies:

structural logic rule	☞	*unary demonstration*

Identity   p → p

```
( ) deep var dvar step occ clar per level
 1   1    2   1    2    1   0    1   1
```

```
(p) p
( ) p
( )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Simplification   (p ∧ q) → p   (p ∧ q) → q

```
( ) deep var dvar step occ clar per level
 4   3    3   2    2    1   1    1   1
```

```
(((p)(q))) p
 ( )(q)    p
 ( )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Addition   p → (p ∨ q)   q → (p ∨ q)

```
( ) deep var dvar step occ clar per level
 1   1    3   2    2    1   0    1   1
```

```
(p) p q
( ) p q
( )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Exportation   (p ∧ q) → r = p → (q → r)

```
( ) deep var dvar step occ clar per level
 4   3    3   3    1    0   1    0   0
```

```
(((p)(q))) r = (p) (q) r
 (p)(q)    r = (p) (q) r
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Resolution   ((p ∨ q) ∧ (¬p ∨ r)) → (q ∨ r)

```
( ) deep var dvar step occ clar per level
 5   4    6   3    3    1   1    3   2
```

```
(((p q)((p) r))) q r
 (p  )((p)  )    q r
 (p  )(     )    q r
     (      )
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

DeMorgan   ¬(p ∨ q) = ¬p ∧ ¬q
             ¬(p ∧ q) = ¬p ∨ ¬q

```
(p q) = (((p))((q)))
(p q) = (  p     q  )
```

```
( ) deep var dvar step occ clar per level
5/4 3/3 2/2 2/2 1/1 0/0 1/1 0/0 0/0
```

```
(((p)(q))) = (p)(q)
  (p)(q)   = (p)(q)
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Biconditional   (p = q) = (p → q) ∧ (q → p)   p = q ↔ (((p) q)((q) p))
                  (p = q) = ¬(p ∨ q) ∨ (p ∧ q)   p = q ↔ (p q) ((p)(q))

```
( ) deep var dvar step occ clar per level
5/4 3/2 4/4 2/2 1/1  –    –    –    –   3
```

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Distribution
p ∨ (q ∧ r) = (p ∨ q) ∧ (p ∨ r)   p ((q)(r)) = ((p q)(p r))
p ∧ (q ∨ r) = (p ∧ q) ∨ (p ∧ r)   ((p)(q r)) = ((p)(q)) ((p)(r))
p → (q ∧ r) = (p → q) ∧ (p → r)   (p) ((q)(r)) = (((p) q)((p) r))
p → (q ∨ r) = (p → q) ∨ (p → r)   (p) q r = (p) q (p) r
                                    (p) q r = (p) q   r

```
( ) deep var dvar step occ clar per level
3/6 2/2 4/4 3/3 0/0  –    –    –    3
2/5 1/3 4/4 3/3 1/0  –    –   1/0   3
```

Figure 7-3: *Unary Demonstration of Structural Rules*
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

`exists`	`(      )`	begin with existence
`create`	`(p) (      )  q`	`create` some pseudo-variables
`reify`	`(p) ((p) q)  q`	`reify` some of the illusory forms
`enfold`	`(((p) ((p) q))) q`	`enfold` some structure

The construction shows why variables within a tautology are pseudo-variables: they are freely created. It also shows that there is no essential difference in unary reduction when applied to atomic variables, to formulas, or to collections of premises. `Reify` assures that the constructed structure remains contextually meaningless. And `enfold` overlays the multiple readings that give symbolic logic its apparent diversity.

Here are three other tautologies from Figure 7-1 constructed using this template:

modus tollens	*disjunctive syllogism*	*hypothetical syllogism*
`(      )`	`(   )`	`(      )`
`q  (      )  (p)`	`q  (   )  p`	`(   q)(      )  (p) r`
`q  (q (p))  (p)`	`q  (q p)  p`	`((p) q)((q) r)  (p) r`
`((((q))(q (p)))) (p)`	`q (((q p)((p))))`	`((((p) q)((q) r))) (p) r`

All of these constructions share a common beginning: `create` and `reify` the desired pseudo-variables. Inference is not a necessity. The uniformity is not surprising since the formalization of logic was built on the syllogisms of ancient Greece. One interpretation of `(A) B` is that `A IMPLIES B`. But to maintain truth, the form of implication originates from an Occlusion, `( ) B`, that can be interpreted as `FALSE IMPLIES B`.[9]

For demonstration purposes we can build a new tautology. The game is to follow the template to construct a unary form without considering its logical interpretation. It's rather tricky not to replicate or embed versions of existing deductive rules. However here's a new rule of deduction:

( )			exists
(( )( )) ( ) p q			create
((p)(q)) (p q) p q			reify
((((p)(q)) (p q))) p q			enfold
Interpreting			*interpret*
(((p)(q)) (p q)) p q	☞	*content* → *context*	IMPLIES
(p)(q) p q p q	☞	(*left* ∧ *right*) → (p ∨ q)	AND, OR
	☞	(¬(p ∧ q) ∧ (p ∨ q)) → (p ∨ q)	NOT-AND, OR
		(p ≠ q) → (p ∨ q)	XOR

The new rule asserts that when two Boolean forms are not equal, one of them is TRUE. The implication is not supported in the opposite direction. That one of p OR q is TRUE does not imply that the other is FALSE.[10] This tautology highlights the controversial ambiguity at the turn of the twentieth century between the interpretation of the word *or* as an inclusive or exclusive connective. Boole, in 1854, modeled Boolean algebra on the algebra of numbers, so that the use of OR in the phrase 4 OR 5 did not include *or possibly both*. In Boole's algebra p OR q was uninterpretable when both were TRUE since Boolean 1 + 1 would equal 2.

7.5 Remarks

The axioms and theorems of propositional logic can be condensed into the three unary logic axioms, each of which deletes structure. The quite limited diversity of Level 3 tautologies suggests, at least through the evolutionary history of formal axioms from about 1870 to the mid-twentieth century, that there was a strong preference for *very simple models* of organized thought. I suspect that the culprit was not an attempt to *be* simple, but rather that what is now seen to be simple was seen to be rather complex during the early evolution of formality. There is certainly evidence that simple approaches to deduction, especially by Peirce, were generally ignored. Seminal technical work is often turbid, needing decades to be refined into clarity. In Chapter 8 we'll see that Cancellation alone is sufficient to axiomatize propositional logic.

cancellation
((A B) (A (B))) = A

Endnotes

1. **opening quote:** L. Wittgenstein (1922) *Tractatus Logico-Philosophicus* proposition 4.462.

2. **than one that is unbounded:** The literals `a` and `(a)` differ only in the viewing perspective of the reader, from the outside or from the inside. From the symbolic perspective `(a)` is the negation of `a`, which appears to carry more import. The two forms are symmetric though. It is historic and cultural influence that makes negation more than a mirror image. Literals are considered to be of equal complexity in circuit design.

3. **automatic discarding of double negations:** Michael Beeson, R. Veroff & L. Wos (2005) Double-negation elimination in some propositional logics. Online 3/24 `https://arxiv.org/abs/cs/0301026` p.7.

4. **variables by operators and relations:** The non-nested shallow version of Pervasion is transcribed here due to the difficulty of expressing the semi-permeability of boundaries in a symbolic notation.

5. **design of silicon circuitry:** Metrics based on the number of logical connectives, variables or wires are no longer a priority for semiconductor design. The industry has moved on to focus on the number of inputs and outputs, the density of wiring, the generation of heat, the power drain on batteries, the capability of extreme miniaturization, and the production yields of silicon chips.

6. **reduce a tautology to ground:** Additional applications indicate embedded lemmas that can substantively impact complexity.

7. **deductive techniques and strategies:** The distinction between computational deduction and cognitive inference has in the past dominated academic discussion. It also reflects the distinction between syntax and semantics, between demonstration and proof, and between artificial and biological intelligence. These dichotomies have been imposed upon symbolic structure and thus on all logical communication. The structural perspective in this volume seeks to avoid this discussion by assuming that any inferential specification that is sufficiently clear enough to be formal also to be computable.

8. **useful rearrangement theorems:** This observation holds as well for cognitive formalization and for the validation of million gate silicon circuits.

9. **interpreted as `FALSE IMPLIES B`:** Figure 8-1 shows that Peirce included this very controversial formula in his 1885 axioms for propositional logic.

10. **that the other is `FALSE`:** Yes, this is a dangerous inference rule since it degrades information.

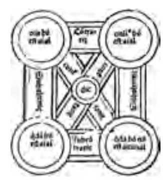
CHAPTER CONTENTS

Chapter 14

Dialects

...it will become increasingly necessary to discover
new languages that intermediate between
the human domain of understanding and
the computational domain of symbolic manipulation.[1]
— L. Kauffman (1997)

Logical connectives have been presumed to be abstract since antiquity; they are the *syncategoremata,* words that refer to nothing yet function to connect words that do have referents.[2] Since logical connectives have no physical meaning, they have been represented by meaningless tokens. However the operations and concepts of logic have a rich diagrammatic history. The images in the margin each show the relational structure of the syllogisms.[3] In Donkey Bridge the syllogisms are razor blades that will dismember all who choose to play rather than study.



Square of Opposition

Geometry of Logic

Donkey Bridge

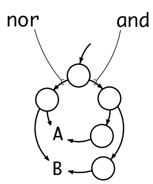
The iconic concept of containment has the same *relational structure* as connection, contact and conveyance. Boundary form can call upon a diversity of **geometric and topological transformations** to indicate change. There are thus a wide variety of *iconic dialects* that can express the same logical intentions.[4]

Symbolic logic is designed primarily for *static* description. Transformation is thus a separate add-on process. Changes are enacted as separate temporal steps. The

t_0 `(a ((a b)(b c)))`
t_1 `(a ((  b)(b c)))`
t_2 `(a ((  b)     ))`
t_3 `(a     b       )`

primary purpose of boundary forms is to manage logic *distinctions*. Like cognitive acts iconic forms are *dynamic*. Iconic dialects are designed to display change through **animation**. Parens forms, for example, simulate animation by unfolding vertically to show change over time.[5]

Iconic form no longer needs incidental symbolic tokens for the *neutral outermost container* `U` or the *explicit void token* `ø`. The background of an iconic image *is* the outermost container. We will at times need to identify its presence and we will still need to provide a global container for those images that are fragmented in the outermost context, but we will rarely need to provide a specific label for the outermost form. The same background is *void*. Similarly we will still need conventions to assert and to maintain the relative absence of form but we will not need to explicitly label non-existence. The innermost container will always either be empty or contain at least one variable.



The two dozen iconic dialects in Figure 14-1 show how quasi-iconic parens can be transformed to generate dialects in the form of enclosures, trees, networks, maps, flows, paths and blocks.[6] Each iconic dialect is illustrated by the same example, the binary exclusive-or, `XOR`. Each image is a composite of the images of two simpler logic functions. `XOR` can be decomposed into the form of `AND`, `((A)(B))` and the form of `NOR (A B)`.[7] Combined by `NOR` they define inequality.

inequality

`A ≠ B` ☞ `((A B)((A)(B)))`

The forms of `XOR` in Figure 14-1 illustrate one-, two- and three-dimensional unary dialects.

— The **parens** dialect is digital and encodable. The language consists of one-dimensional delimiters in fractured bracketing relationships with one another.

— The **enclosure** dialects show two-dimensional containment. The language consists of nesting relationships among closed curves.

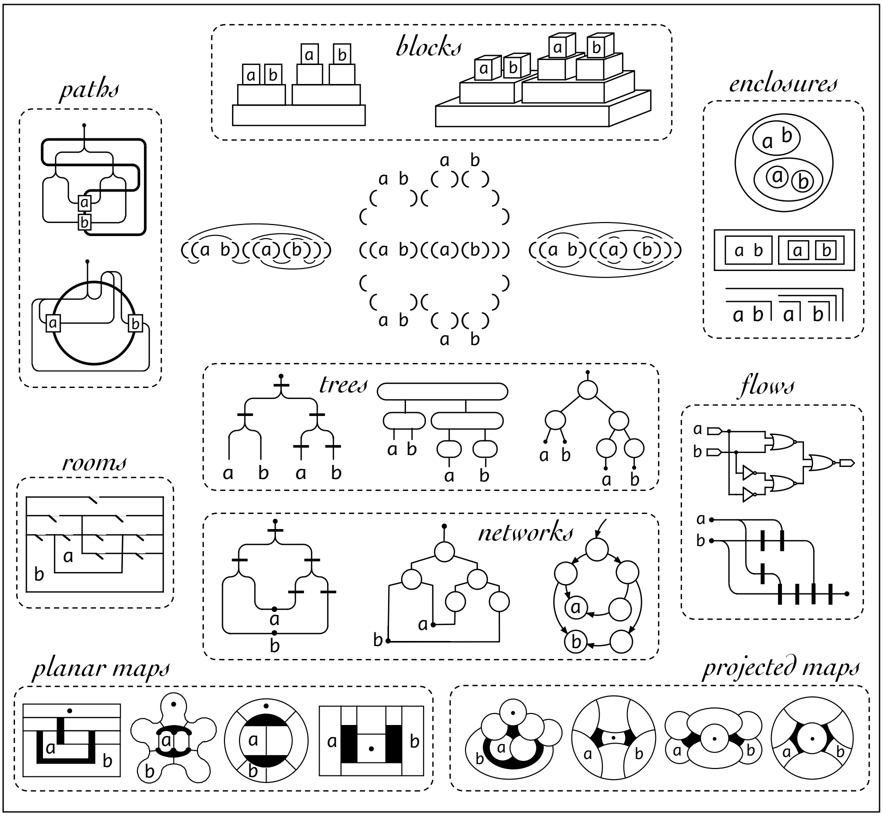


Figure 14-1: *A Selection of Unary Logic Dialects*

— The **tree** dialects illustrate containment relations between input and output. The language consists of two-dimensional tree graphs.

— The **network** dialects are traversable acyclic graphs displayed in two or three dimensions. The language consists of nodes as containers and links as containment relations.

— The **flow** dialects indicate flow from input to output passing through minimal processors. The language is paths obstructed by functions that combine and change their local inputs.

- The **planar map** dialects are traversable territories. The language consists of areas with shared borders and an explicit *you-are-here* starting point.
- The **projected map** dialects use projected three-dimensional object occlusion to simulate nesting by depth. The language is territorial maps with borders that indicate nesting.
- The **room** dialect is a three-dimensional environment that a participant can explore. The language is arrangements of rooms and doors.
- The **path** dialects show traversable paths that cross over a single boundary. The language consists of directional paths that diverge to cross the same boundary many times.
- The **block** dialects are manipulable. The three-dimensional language consists of physical objects in stacking relationships with one another.

The conceptual and computational features of unary logic are carried into these iconic forms. The first section in this chapter describes **dnets**, the iconic dialect that has proven to be the most useful for modeling huge logic networks and for asynchronous parallel computation. The following sections then describe the other iconic dialects in Figure 14-1. Each dialect is constructed from the parens notation. The unary axioms are then expressed in the new dialect and used to demonstrate Detachment and Absorption.

14.1 Distinction Networks

A binary relation can be modeled by a *graph,* a collection of nodes and a collection of links that connect pairs of nodes.[8] A distinction network is a **directed acyclic graph** with nodes that represent containers and directional links that represent the `contains` relation. Nesting is indicated by an arrow. An *acyclic graph* does not have loops or cycles. A directed acyclic graph is called a DAG by computer scientists and a *rooted tree* by mathematicians. Containment is not a partial ordering since it is neither reflexive nor antisymmetric nor transitive.

`1 contains 2`

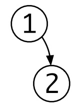
☞

`(1,2)`

☞

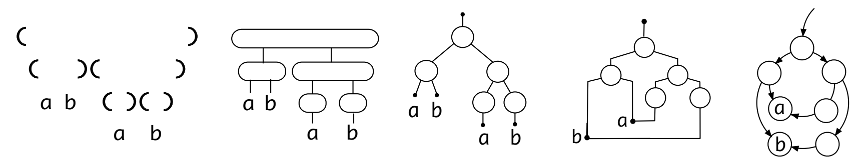
$(2)_1$

☞



Dnets are a visual display of *ordered pairs*. The first label of each pair identifies a container and the second label identifies a form that is contained. Dnets do not require the symbolic null token `ø`. A pair such as `(A,ø)` would identify an exiting arrow with no connection. Instead, the deepest leaf nodes in a dnet are either marks or variables. Nodes with variable labels can stand in place of arbitrary subnetworks.

Here is the conversion of parens forms to dnets for logical `XOR`: `((a b)((a)(b)))`.

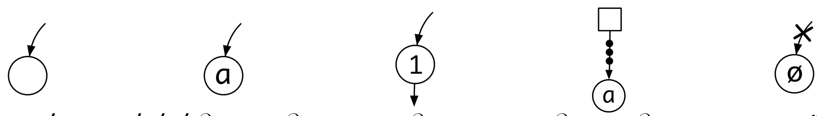


extruded parens *closed parens* *tree* *network* *dnet*

The tree form of a DAG includes separate nodes for replica variables. The network form combines replicas into a single node with multiple upper links. The final circular dnet graphically shows a connectivity subnetwork that occurs frequently in logic formulas. The five nodes of the innermost loop are the parens form `((a b)((a)(b)))`, the three argument logic formula `ITE a b ¬b` that equals `a XOR b`.

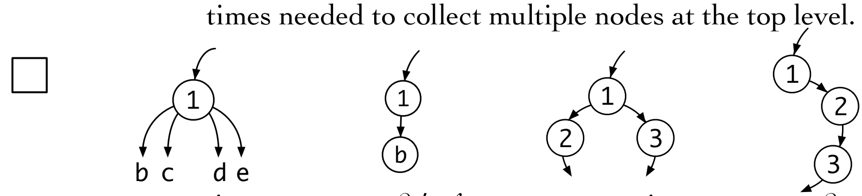
Structure

A **form** is a directed network of connectivity. A **subform** is the subnetwork of any given node. *All nodes are roots* for their contents. Each subnetwork is independent. In an **acyclic network** two containers cannot mutually contain each other either directly or indirectly. A node with no upper links is the outermost container, the **root** of a rooted graph. An node with no lower links is a **leaf** of a rooted graph. A leaf with no label is the empty container *mark*, Here are some common examples of dnet nodes:



mark	*labeled ground*	*node*	*deep node*	*no void*
( )	a	$(\ldots)_1$	⁝ a ⁝	

The connectivity of network nodes is displayed visually by links. Multiple downward links identify multiple contents of a container. Multiple upwards links identify forms replicated in multiple different containers. Here are some types of node connectivity. The square representing the root is not a node, it is the outermost nested-pair sometimes needed to collect multiple nodes at the top level.

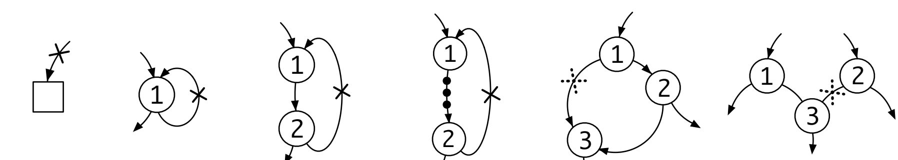


root	*generic*	*named leaf*	*same container*	*nested*
$[\ ]_U$	$(b\ c\ d\ e)_1$	$(b)_1$	$((\ldots)_2(\ldots)_3)_1$	$(((\ldots)_3)_2)_1$

The infrastructure of network connectivity replaces both of the abstract constructs that hold symbolic collections together: **sets** (i.e. unordered collections of ordered pairs separated by commas) and **predicate calculus** (i.e. collections of binary relations joined by conjunction).

Constraints

Dnets make the shift from physicality to abstraction visually clear. The heavy X across a link indicates that the link is not permitted. The dashed X indicates that the link is permitted but is not asserted by rule.

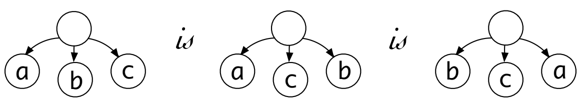


outermost	*irreflexive*	*asymmetric*	*acyclic*	*not transitive*	*physical*
$\underset{x}{(}[\ ]_U\underset{x}{)}$	$\underset{x}{(}1\ldots)_1$	$\underset{x}{(}(1\ldots)_2)_1$	$(\ldots ⁝ \underset{x}{(}1\ldots)_2 ⁝)_1$	$(\underset{x}{3}\ (3\ldots)_2)_1$	$(3\ldots)_1\underset{x}{(}3\ldots)_2$

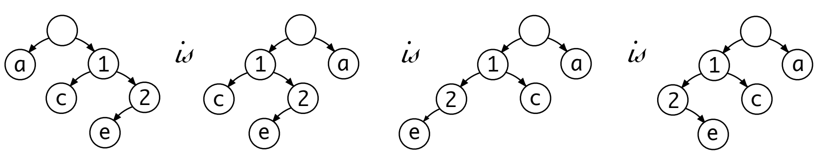
Like physical containment, links are *not* reflexive, symmetric or cyclic. When a node has more than one upper container it is no longer a physical model since we cannot place the same physical container in two different places. Multiple upper links occur whenever a tree is converted into a network by merging *replicas* of variables.[9]

Flexibility

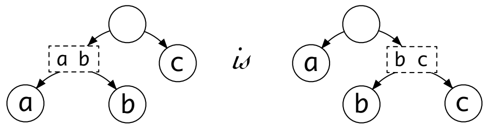
Linear constraints are *implementation details* rather than fundamental properties. Conceptually, all nodes can be processed, traversed, and/or viewed at the same time. Rooted trees and graphs do not have a preferential ordering of their branches. Ordering of content nodes within a dnet is not even possible since there is no structural information within the collection of links to support an ordering. The visual image of equal orderings below *distorts* the iconic perspective in favor of showing linear permutations to be equivalent.



Dnets, in contrast, incorporate the concept of perspective, so that *commutativity* does not require restructuring the representation. In the freedom of three dimensions, spatial transformation can be achieved by a greater diversity of methods, including two-dimensional spatial reflection, temporal network traversal, and three-dimensional rotation. By permitting a three-dimensional reading, absence of ordering can be expressed equivalently as relocation of a view-point.



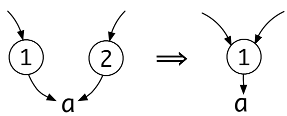
An *associative* symbolic function is also a temporal sequence of accumulation. In a network accumulation over time can be represented by nodes that create additional depth in the representation.



From the perspective of dnet structural connectivity, grouping the contents of a container (i.e. establishing an arity) violates the independence of those contents. Since containment itself is the iconic grouping operation, support of *binary* arity would require the insertion of additional containers, in effect the creation of two different types of containment.

Structure Merging

Structure merging is the process of using the same label to identify the same subnetwork in different locations within a larger network. Merged structure in dnets maintains the same network connectivity.



structure merging a literal

$$(a)_1(a)_2 \quad \Rightarrow \quad (a)_1$$

Symbolic unification is usually goal-directed, matching patterns, for example, to move deduction forward. In almost all descriptions of symbolic transformation, structure merging is not mentioned; instead replication and labeling of structure is considered to be free. In dnets structure merging is an optimization technique to reduce redundancy and is not strictly necessary for axiomatic reduction. Condensing identical structures is necessary for large logic networks, for example, in optimizing the routing of silicon circuits.

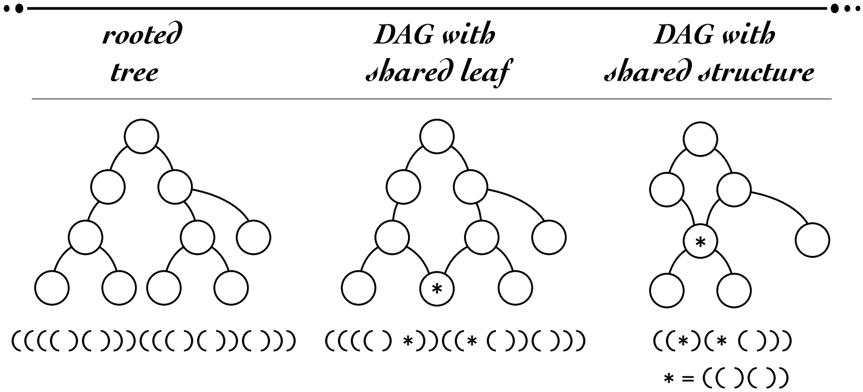


Figure 14-2: *Structure Merging Trees into Directed Acyclic Graphs*

The simplest case for dnets is merging marks to implement the axiom of Void Calling.[10]

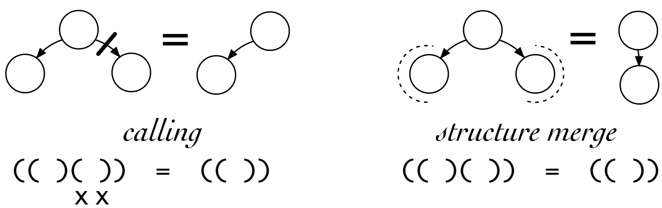
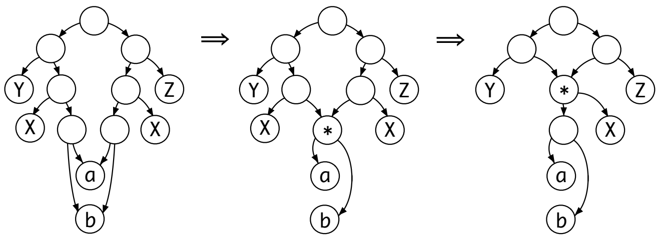


Figure 14-2 shows the conversion of rooted trees to directed acyclic graphs through structure merging unary arithmetic forms. Merging potentially reduces multiple applications of a transformation during reduction. Within symbolic systems Distribution validates merging but only within tightly constrained patterns such as the distribution of multiplication over addition. Algebraic factoring is another application of constrained merging. Unary logic bypasses these constraints by incorporating only one type of relation.

((a b)(a c)) → a ((b)(c))

(a*b)+(a*c) → a*(b+c)

Structure merging in dnets is a bottom-up process beginning with merging nodes with the same label, i.e. the same named variable. Larger identical subnets can also merge iteratively. Here is an example, the merged nodes are indicated by an asterisk.



Like other symbolic notations structure merging in parens forms is achieved by naming a given structure and then substituting the name for the structure throughout a form. In the above example

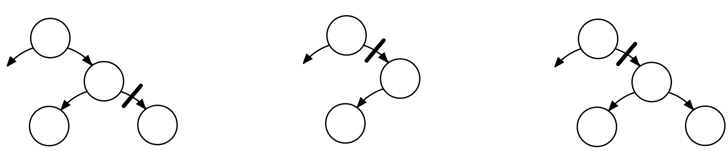
A = ((Y (X (a b))) (Z (X (a b))))
    let P = (X (a b)), *substitute* P *for* (X (a b)) *in* A
A = ((Y P) (Z P))

Multiple references within a dnet are represented by multiple upper links rather than by multiple occurrences of the same label. Merging facilitates both pattern-matching and memory management.[11]

Asynchronous Arithmetic of Form

From the perspective of computer architecture, a dnet is a fine-grain massively parallel processor.[12] All nodes in a dnet are continuously active and responsive. Dnet transformations are entirely local, each node signals only its direct neighbors. There is *no global coordination*. Distinction networks generate change similar to a *cellular automata,* by individual nodes assessing only the existence of their direct neighbors.

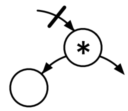
Figure 14-3 briefly illustrates the dnet signaling regime for unary arithmetic. Crossing and calling are disconnections between nodes.



calling	*crossing*	*void calling*
(( )( )) = (( ))	(( )) =	(( )( )) =

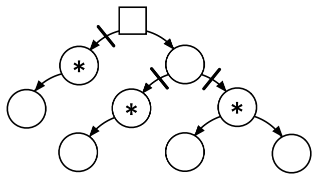
OCCLUSION

`if no lower links,`
`delete your container(s)`



Trigger: ¬∃ `lower links`
Action: `send upper nodes DISCONNECT`

Trigger: `receive DISCONNECT signal`
Action: `disconnect upper links`



example [(( ))((( ))(( )( )))] ⇒ [( )]

Figure 14-3: *Asynchronous Unary Arithmetic Reduction*

Each node has internal access to the *presence* of upper and lower links and can assess when there are connected no lower links. Links are not counted, quantification is purely existential. Whenever a node has no lower connections (i.e. it is a mark) the node sends a `DISCONNECT` signal to all of its upper links (i.e. its containers). Whenever a node receives a `DISCONNECT` signal, the receiving node deletes its upper links, in effect disconnecting itself and its subnetwork from the remaining nodes in the network.

Nodes are pruned until the passive outermost root has either no lower links, which means the dnet no longer exists (what we have been calling *void*), or it receives a `DISCONNECT` signal, which means only one node exists (what we have been calling *mark*).

For the example in Figure 14-3 the four nodes with no lower links concurrently send their upper containers (indicated by an asterisk) the `DISCONNECT` signal. The three upper nodes concurrently disconnect from their upper nodes, leaving the reduced arithmetic form, a single mark.[13]

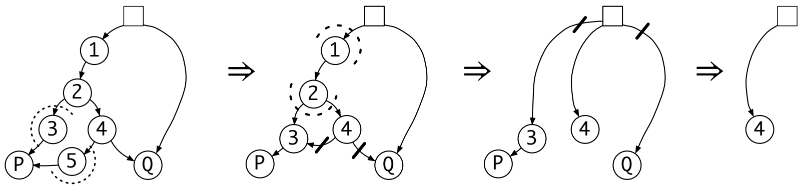


Figure 14-4: *Parallel Dnet Demonstration of Detachment*

Asynchronous Computation

Figure 14-4 illustrates the parallel reduction of the dnet for Detachment. Multiple applications of the axioms occur dynamically throughout the dnet. Nodes 3 and 5 first merge, which then allows three concurrent deletions of connectivity, two by Pervasion and one by Involution. Occlusion then completes the demonstration.

Asynchronous computation does not imply the absence of sequential steps. There are necessary parallel and sequential regimes within a reduction as has been demonstrated in earlier chapters. The primary feature of asynchronous processing is the absence of a global control structure.

Figure14-5 presents a slightly more complex example of the reduction of an algebraic dnet. The intent is to provide a visual comparison of the same reductive sequence in the annotated parens form and in the dnet form. This example is the deductive reasoning problem from Chapter 10.4 about eating fruit and vegetables. Images 1 and 2 are the only cases of parallel reductions. Images 2 and 4 illustrate the differences between textual labeling and structure merging. Deletion of links in images 2, 4 and 6 prepare the dnet for merging in 3, clarifying in 5, and occluding in 7.

premise	*premise*	*premise*	*conclusion*
$((a)_6\ p\ g)_1$	$((g)_7(m\ (k)_9)_8\ p)_2$	$(((b)_{11}\ p)_{10})_3$	$(a)_4(m)_5$

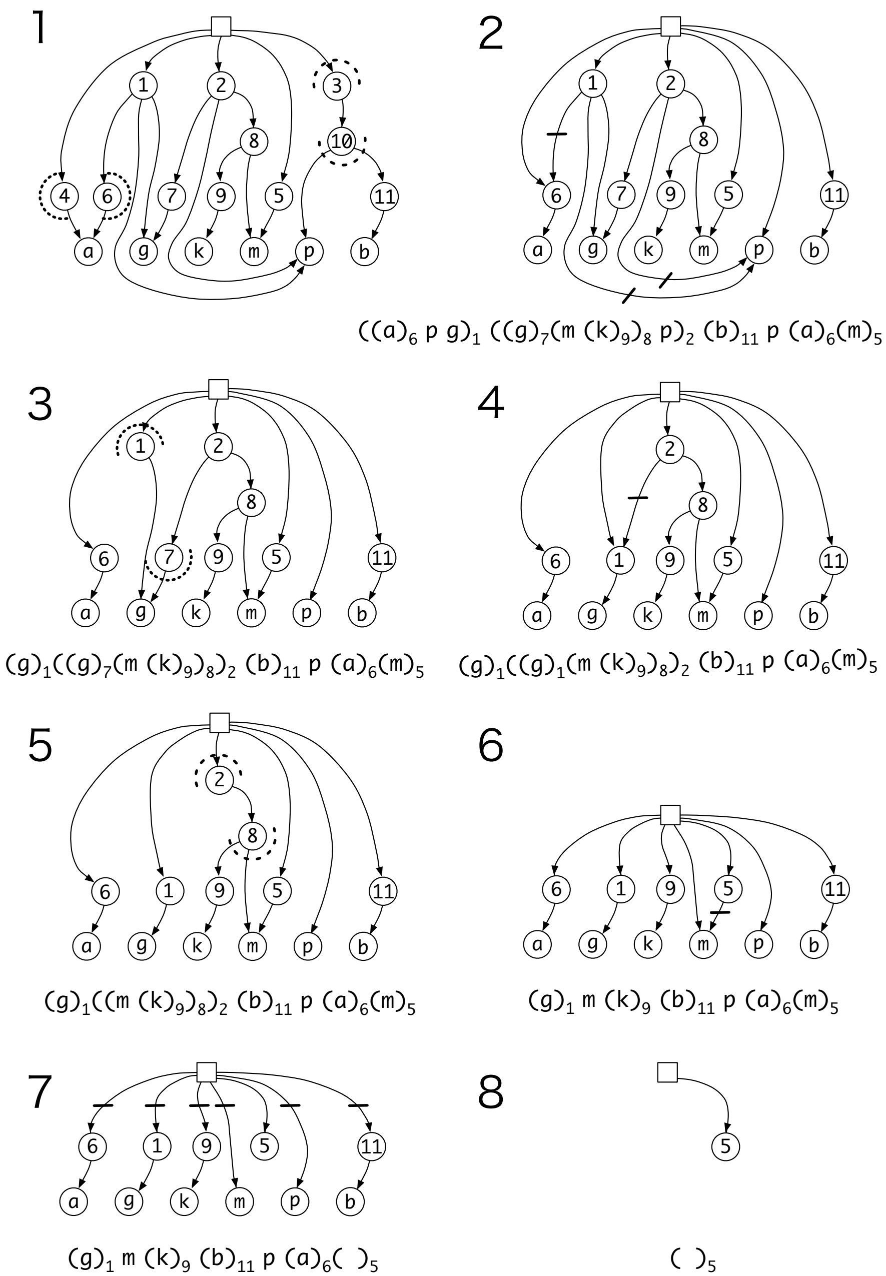
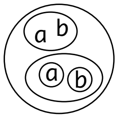
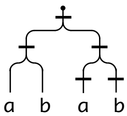


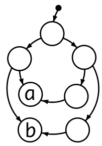
Figure 14-5: *Annotated Parens and Dnet Reduction*



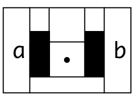
enclosed circles



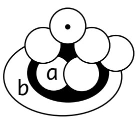
crossbar tree



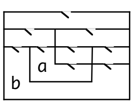
distinction network



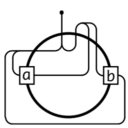
planar map



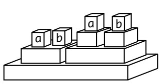
stepping stones



doors and rooms



capform path



stacked blocks

The figure is deceptive in its inability to represent asynchronous actions. It is the signal delay between nodes that would affect different reduction paths. These delays are physical and in an asynchronous regime *not managed*. In conventional hardware a global clock keeps the concurrent actions of all transitions in lockstep, but clocking is not available across distributed computational resources. As well, different transformations engage in temporally different signaling protocols. An application of Pervasion, for example, takes longer when the replica is further away from the host.

Parallel processing is broader than simply a more efficient method of computation. Parallelism frames logic as a **dynamic interactive system**. The change in perspective is from symbolic representation coupled with independent transformations to dynamic concurrent activity choreographed by local signaling that leads to *global optimization achieved by local deletions*.[14] The outside viewing perspective is no longer a forced choice, asynchronous transformation has no privileged external perspective.

Symbolic formal mathematics preserves the conceptual and structural perspectives of a pre-computational era, a perspective that is not suitable to describe formal iconic processes. There is no convenient symbolic notation for *structure merging*, indeed replication is the primary method that symbolic notations coordinate their structure. Local concurrent decision making in dnets coupled with deletion actions can be *described* by sets of ordered pairs and by conjunctions of relations, but the operations associated with sets and logic are not employed and transformations are not choreographed.

14.2 Iconic Dialects

In this section the dialects in Figure 14-1 are described in more detail. The dialects are loosely organized into structural families. A selected dialect from each family is constructed from parens forms. The eight selected dialects are shown in the margin. The unary logic axioms in each dialect are illustrated and then used to demonstrate Detachment.

$$(P \wedge (P \rightarrow Q)) \rightarrow Q \quad ☞ \quad (((P)_3((P)_5\ Q)_4)_2)_1\ Q$$

detachment

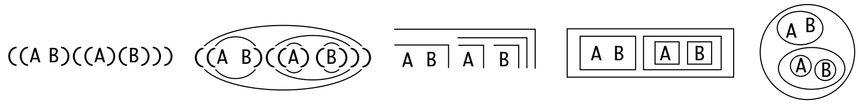
Figure 13-5 includes a unary logic demonstration of Detachment for ordered pairs. To reduce complexity and to improve readability, deep boundary semipermeability has been simplified into the single semipermeable boundary of Shallow Pervasion.

shallow pervasion
(A (A B)) = (A (B))

Different dialects elicit different reduction sequences. Just like symbolic notations that vary over relational and functional perspectives, each of the iconic dialects suggests its own *method of logical thought*. Some are more useful than others, of course. From the perspective of the symbolic syntax/semantics barrier, the iconic dialects are different *interpretations* rather than different notations. However, from the perspective of iconic form, the dialects extend the meaning of logic formulas with new tools for computation and for rational thinking. Iconic dialects allow us to engage in physical and participatory exploration of the processes of deduction.[15] Several of these dialects become too clumsy for theorems like Arrangement and Pivot, they are included specifically to suggest different cognitive models.

The images in this section use a large dot, • , to identify a **view-point** or a starting location when the outermost container is not apparent. Several dialects encourage participation from the inside rather than reading from the outside. Technically the view-point is from the metalanguage and does not interact with the transformation sequences. For several dialects it is superseded by alternative participatory cues such as the foreground/background cue for projected maps and the gravitational top-to-bottom cue in graphs. In several other dialects the view-point is used redundantly to enhance readability.

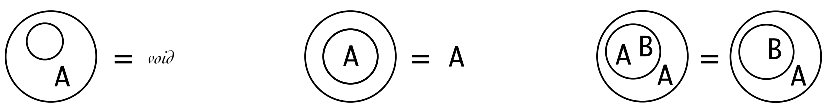
Enclosure Dialects



parens *enclosed parens* *crosses* *boxes* *circles*

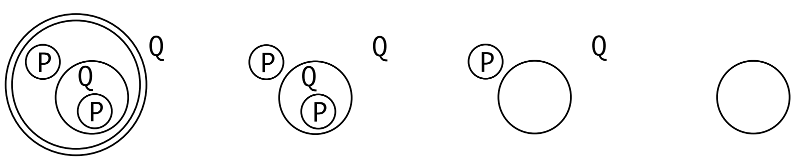
In Figure 14-1 parens boundaries are enclosed above and below to generate the more familiar enclosure dialects. Extension of parens forms into two dimensions removes the incidental ordering and fragmentation imposed by the typographic representation.

Spencer-Brown's cross notation bears a resemblance to parens in delineating only two sides of implicitly surrounding containers. Peirce's Existential Graphs use the circular enclosure dialect. Kauffman's iconic solution to the Robbins problem in Chapter 11.5 uses the rectangular box dialect. Enclosure dialects visually achieve Spencer-Brown's definition of distinction as *perfect continence*, or rigorous constraint.



occlusion *involution* *shallow pervasion*

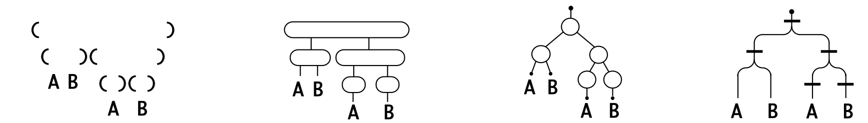
The three unary logic axioms are displayed for Peirce's **circle dialect**.[16] Capital letters stand in place of any containment configuration. Below, these axioms demonstrate the form of Detachment.



detachment *clarify* *pervade* (P) Q *occlude*

This is a very common reduction sequence. Nested-pairs are clarified, subforms are pervaded and Occlusion terminates. A slightly different demonstration of Detachment augmented by parallelism is included in Chapter 1.5 as a visual introduction to unary deduction. Enclosure dialects provide strong visual support for pattern-recognition of the reduction axioms.

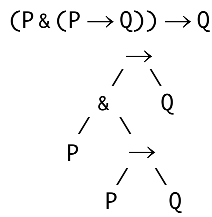
Tree Dialects

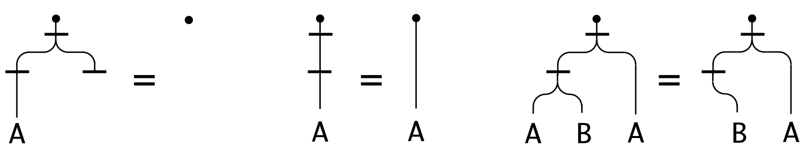


extruded parens *capped parens* *tree graph* *capform tree*

Extruding a parens form downward creates the tree and graph dialects. The view-point identifies the root of the tree. Labels identify the leaves, which can be arbitrary subtrees. Due to strong parallelism structural reductions can be applied to any section of a tree or graph.

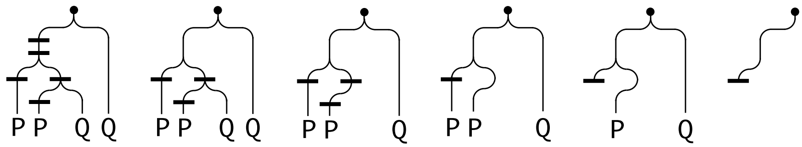
Trees are closely aligned with textual languages. Any string of text can be extruded to create a tree. A logic formula for example is a binary tree mixed in with unary negation nodes, as illustrated by the tree graph. Trees have been widely studied in computer science since they are the structure of documents stored within a file system, the stack processing of software programs, and the many ways to optimize both storage and processing in both software and hardware. *Semantic tableaux* use trees of deductive steps to display the structure of a proof. Still, trees are *quasi-iconic* due to their incorporation of symbolic replicas.





occlusion *involution* *shallow pervasion*

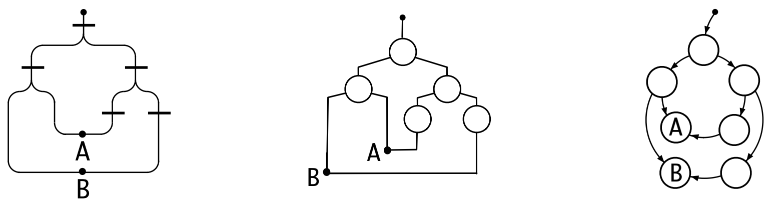
The unary logic axioms are shown for the **capform tree dialect**. In circuitry the bars are `NOT` gates that change the polarity of current between `ON` and `OFF`, while the merging links represent a `WIRE-OR`.[17] In Occlusion the view-point is utilized to locate the reader's perspective. It identifies an outermost boundary. Tree forms are not particularly useful for boundary transformations, however they do provide a bridge to the graph dialects.



detachment *clarify* *pervade* Q *clarify* *pervade* P *occlude*

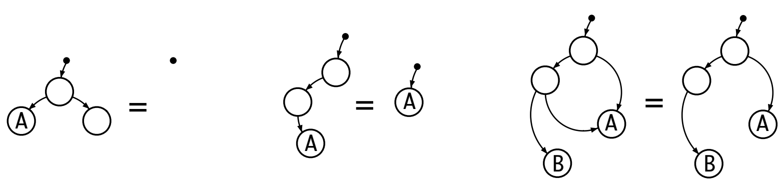
The demonstration of Detachment in the capform tree dialect is shown directly above.

Network Dialects



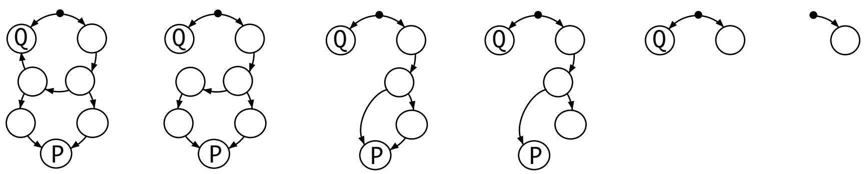
capform network *NOR circuit* *distinction network*

A tree is converted into a network by merging named leaf nodes to construct a directed acyclic graph that supports multiple reference with multiple links rather than with multiple labeled nodes. Networks then support further structure merging, of which there is none available in XOR. The capform network simply merges leaf nodes of the capform tree. The circular nodes in the NOR circuit can be read as NOR gates, converting the NOT bar and the WIRE-OR into a homogeneous circuit. The distinction nodes accommodate the former NOT gates by not having an arity.



occlusion *involution* *shallow pervasion*

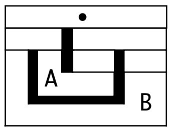
Two conventions identify the outermost container in **distinction networks**. A gravitational orientation places the root node towards the top of the page, making it the "top" node. This convention requires meta-information about orientation in space and is thus a viewing perspective as well as a design choice. An alternative technique is to extend a link out of the graph while locating the context explicitly with a view-point. When describing graphs as processing networks, variables are inputs and the root node is output. Dnets are reversible since the reduction rules can be implemented in either direction. Although trees are displayed in two-dimensions, display of networks requires three-dimensions, since a planar display with multiple links to a leaf node generally creates the incidental artifact of links that cross in a projected two-dimensional image.



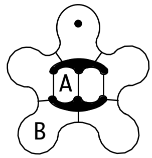
detachment *pervade* Q *clarify* *pervade* P *occlude* *occlude*

Unlike Figure 14-4 this demonstration of Detachment incorporates neither parallelism nor structure merging. The network does represent asynchronous processes. The execution order of these transformations is indeterminate, determined by local physical processes rather than by design. The directional arrows in network dialects make recognizing the patterns of the unary axioms particularly easy. Deep Pervasion identifies *paths* within the network and is implemented by deletion of the last link of longer paths whenever multiple paths converge to the same node. Although Occlusion and Involution are purely local, Pervasion utilizes the entire network. Initiating a path search and implementing a successful path search are both local actions. No node needs to know more than the presence of its local connectivity.

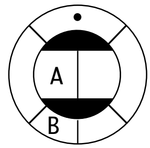
Planar Map Dialects



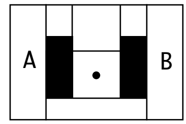
rectangle map



blobby map



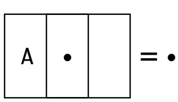
circle map



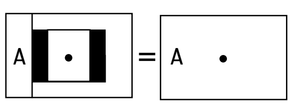
centered map

A distinction network can be converted into a map by enlarging each node until neighbors touch, constructing a common border in place of a common link. Network nodes become map territories while containment links become territorial borders. Maps invite a network perspective; there is only one territory for each variable. However maps tend to lose the directionality of dnets. Planar maps require a view-point to identify the outermost container and to orient nesting. The view-point serves the same purpose as the outermost container in parens notation.

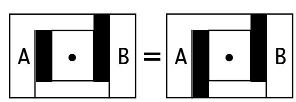
Above, the different shapes of territories create an apparent diversity of planar maps. Mapping onto a plane flattens perspective, just like mapping into a line flattens access. Euclid's plane geometry provides an inside but it allows neither lines to cross without intersection nor boundaries to obscure. Nesting in planar maps becomes the sequential crossing of borders. Non-territorial spaces that are captured by surrounding territories, here indicated in black, are also needed to maintain the separation of borders as links.



occlusion

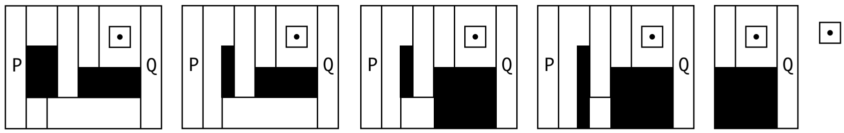


involution



shallow pervasion

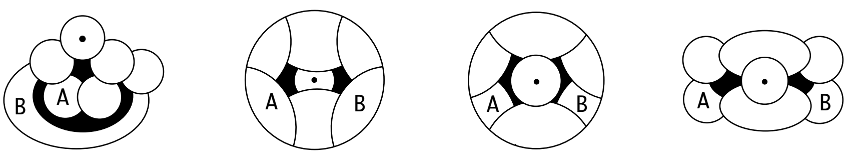
These are the unary axioms for the **centered planar map** dialect. Since the outermost territory is no longer connected to the exterior space of the image, explicit outermost containers are necessary. The view-point serves to identify the outermost territory. For Involution the deletion of the two empty territories leaves the territory of the label A together with the view-point that indicates that A is outermost.



detachment *merge* (P) *pervade* Q *pervade* (P) *occlude twice*

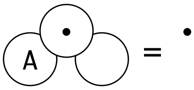
In this dialect structure merging seems easy and does not cause confusion. Taking separate paths through territories makes Pervasion more difficult to see. Where to delete the former borders is error prone. Determining if territorial borders `clarify` is tricky. Following nesting paths is not at all obvious. Managing connectivity by changing the size of non-participating black areas is unintuitive. A global perspective helps to manage these difficulties.

Projected Map Dialects

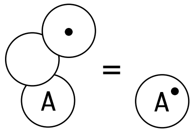


stepping stones *cave* *centered cave* *centered steps*

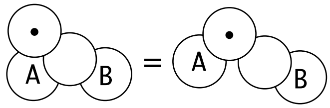
Projected maps can exhibit nesting by overlap, a visual simulation of a three dimensional perspective. The curvature of common borders in the stepping stone dialect incorporates visual cues for a third dimension that locate the top step and show the relative nesting depths of other steps. A redundant view-point is added to clearly identify the top step. The stepping stone dialect resembles a dnet until the multiple links to B enforce a spatial distortion.



occlusion

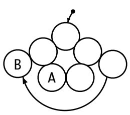


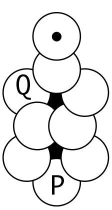
involution



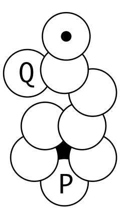
shallow pervasion

As indicated by the reduction axioms the **stepping stones** dialect does pose some difficulties. The labeled and unlabeled steps do not create excess complexity because their behavior during reduction is the same. Labels identify only position in the stack. A global context is needed when there are steps that are labeled, steps that are not labeled and non-step separator spaces. The essential weakness is that steps with more than one lower neighbor require more effort for pattern-matching. With structure merging, arranging common borders is awkward. Shallow Pervasion requires the `A` step to be at two different apparent levels, a poor idea for physical steps. Pervasion of `A` changes the depth as well as the location of the `A` step. The same unevenness of visual depth also shows up in the centered steps dialect. Slight design difficulties in simple forms magnify in more complex forms. In the stepping stone `XOR` form, for example, the shape of step `B` must be distorted. A leaf step with several container steps makes the projection of map connectivity quite impractical.

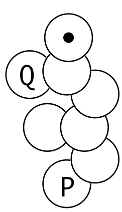




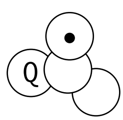
detachment



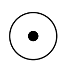
pervade `Q`



pervade `(P)`



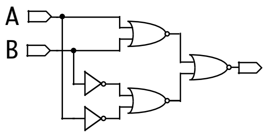
occlude



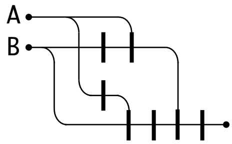
occlude

In the above stepping stone demonstration of Detachment the awkward configuration of steps is avoided by not clarifying. The extra steps avoid cliffs. Enclosure dialects strongly encourage `clarify` reductions while map dialects do not.

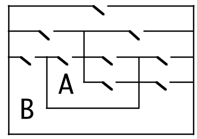
Flow Dialects



silicon circuit

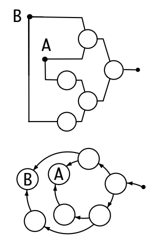


crossbar network



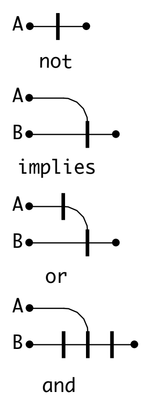
rooms

These three dialects form a miscellaneous category, created to include two historical dialects and one experiential derivative of projected maps.



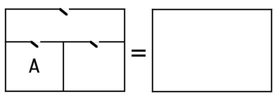
Silicon circuits are of course logic networks conceptually described and designed using classical propositional logic. The particular layout of silicon `XOR` above is composed of `NOR` gates which map to the logical interpretation of distinction nodes in a dnet. An `XOR` graph network and an `XOR` dnet are turned sideways in the margin for comparison to the silicon circuit image. They are graphic variants of the same logic functionality, with the exception that dnets have a fundamentally different behavioral structure inside each node. So do silicon circuits for that matter, the image above is an abstraction at the conceptual level. The inside of a silicon "logic" gate is a network of transistors that have an output that would be expected by symbolic logic but an interior electrical mechanism that is very different from a Boolean look-up table.[18]

a version of Frege's iconic logic

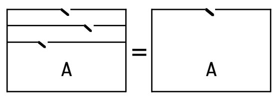


The **crossbar network** is fundamentally different than the other networks in this chapter. It is a reformatting of Frege's 1879 *concept writing* iconic dialect that provided a basis for the first formalization of predicate calculus.[19] Propositions flow through the network from left to right, half a century prior to the invention of the silicon circuit. Bars negate a single input. Two-input bars negate the upper input and then join the two inputs by disjunction, as if a wire `OR`. Two-input bars thus enact material implication, the upper input implies the center input. With practice it is relatively easy to read crossbar networks as parens forms.

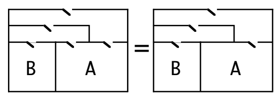
The **room dialect** is a planar map with directional borders indicated by doors. Rooms are also an example of an *experiential logic*. As well as taking the objective view "from above", we can imagine walking through the rooms as if inside a building, selectively closing irrelevant doors. The *void* result is when there are no inward opening doors from the global outside. The marked result is finding an unlabeled room with a single door that opens inward from the outside. Here are the axioms:



occlusion



involution



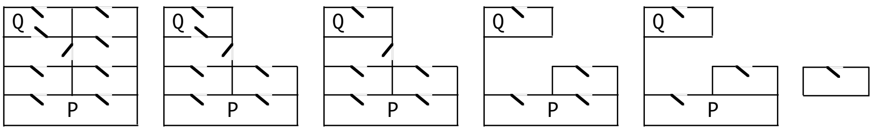
shallow pervasion

Rather than enforcing separation of territories with non-territorial black space, the walls of rooms enforce non-adjacency, while open doors represent shared borders. Occlusion closes off several rooms, creating an irrelevant room by denying access. Involution deletes walls or removes doors to make a hall. Pervasion closes doors that permit taking a longer route to the occupied room. Longer paths are defined by going through more doors.

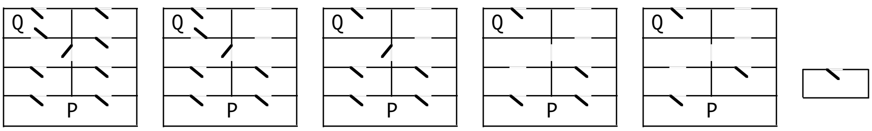
Subjective exploration does not require a global perspective, analogous to the asynchronous reduction of distinction networks. An explorer can only enter a room when a door opens inwardly, and can leave a room through any open door. Possible instructions to achieve axiomatic reduction from the inside might be:

- *Occlusion:* If the room you are in provides access to an unnamed room with no other doors, exit the room you are in and close the door.
- *Involution:* If the room you are in has only one other door and it leads to another room with only one door, then remove both doors.
- *Pervasion:* If the room you are in leads to a named room with other doors, and you can reach the same room by a longer path, close the door in the named room that leads to the longer path.

The success criteria is to find an empty room with a single door to the outside. Here is the search strategy applied to detachment. The upper demonstration restructures the building, an external perspective. The lower demonstration is the same sequence expressed by closing and removing doors.

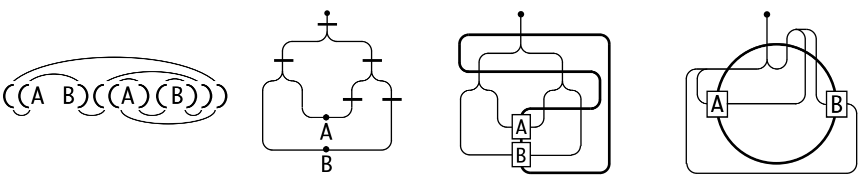


detachment *clarify* *pervade* Q *clarify* *pervade* P *occlude*



The experiential perspective supports the development of experimental theories, aka deductive strategies. An explorer could, for example, try to close as many doors as possible first, in effect implementing Deep Pervasion. For Detachment this strategy ends with two applications of Occlusion. Locating the pervaded doors first leads to rapid termination of the search.[20]

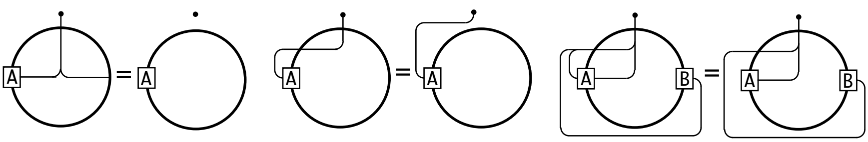
Path Dialects



capform parens *capform network* *merged distinctions* *capform path*

Path dialects were inspired by Louis Kauffman's *capform* representation that connects *opposite sides* of a parens form, as compared to connecting matching sides for the enclosure family.[21] The above series shows the construction of the path dialect form of XOR.

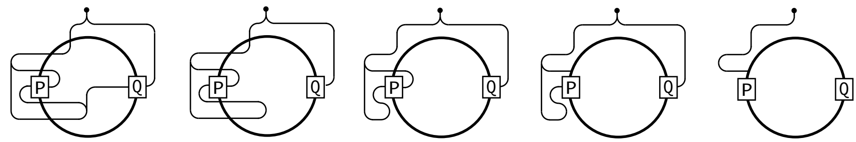
The capform itself is a *single* container distorted by the layout of parens boundaries, with some labels on the inside and some on the outside. Binate variables will have labels on both sides of the container. The second image is a capform network, with nodes represented by bars, and links creating a network of paths from the outermost view-point to each variable leaf. In the third image a single loop is drawn through all capform bars.[22] The merged nodes of the capform network *is* the capform distinction. The containment links are not changed, they are now a network of paths crossing over a single **universal distinction**. Merging replicas not only applies to labels, capstone paths merge containers as well. Finally, the network of connections between capform bars is distorted to accommodate rendering the universal distinction as a circle.



occlusion *involution* *shallow pervasion*

The universal distinction separates even depths of nesting on the outside and odd depths of nesting on the inside. The distinction itself becomes semipermeable for Pervasion on both sides, determined solely by the parity of intervening crossings between view-point and variables. The path dialect trades replication for sequence. To reduce a capform path you follow a path to identify axiomatic reduction patterns. Occlusion is triggered whenever a path terminates at the universal distinction. The path that leads to the termination is deleted, together with all branches along that path. When a path crosses the universal distinction and then recrosses, without any intervening branches, the path can be modified by removing the two crossings by Involution. And when there are multiple paths leading to a variable, longer paths can be deleted by Pervasion. The shortest path from a host to a replica makes all other paths to the replica illusory.

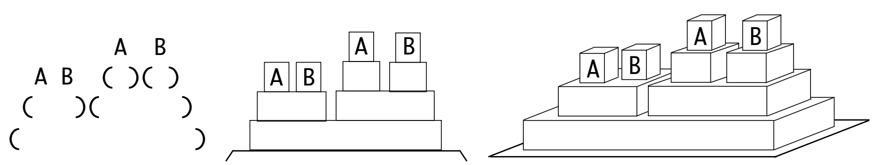
The path dialect is experiential from both the objective outside and the subjective inside. We can follow paths "from above" to identify patterns that support path simplification. Paths are also scalable for physical exploration as literal paths crossing and recrossing the same fence or river.



detachment *pervade* Q *clarify twice* *pervade* P *occlude*

The path dialect elicits significantly different cognitive and computational perspectives. The node/link model of dnets first changes into the barrier/path model of capform networks. Capform bars are a passive communication interface, while the path represents the dynamics of the evaluation of the form. Nodes in a dnet connect and functionally transform convergent links, while the convergence of paths in a capform network is independent of the bars. In the path model with a universal distinction, the capform bars have become a global interface mediating, indeed defining, the evaluation process. Paths have returned to a passive role. A semiconductor design analogy is that dnets represent a circuit schematic with nodes as logic functions and directional links as input and output, while capform networks represent the transistor-level model with bars as inverting transistors and paths implementing disjunctive logic through physical wiring.

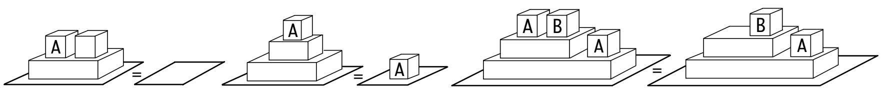
Block Dialects



extruded parens *stacked rectangles* *stacked physical blocks*

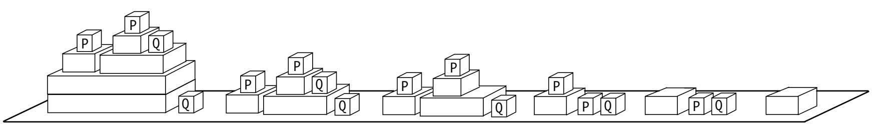
Block dialects are constructed first by extruding a parens form upward and capping each parens boundary to form a rectangular enclosure. Variable labels too are given their own rectangular enclosure. Rectangles are stacked so that containment is replaced by contact. The two dimensional rectangle stack is then extruded into a third dimension. The inert outermost block in a stack is the table. An empty table is *void*.

The intent is to construct physical blocks under the influence of gravity so that containment becomes `is-supported-by`. Block dialect forms are physically manipulable. Reduction rules identify configurations of stacks from which blocks can be removed.



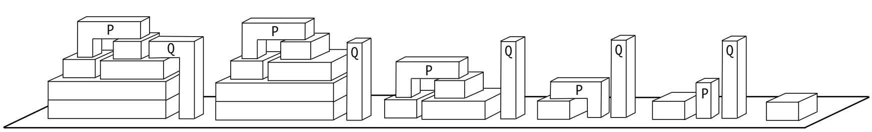
occlusion *involution* *shallow pervasion*

The demonstration of Detachment is intended to be through physical manipulation of concrete objects standing in place of abstract logical forms.



detachment *clarify* *pervade* `Q` *clarify* *pervade* `P` *occlude*

Block forms are quasi-iconic since the named blocks are not merged. It is possible to create arches that unite replica blocks into one form. This, however, degrades physical accessibility to construction and manipulation. Here is the demonstration of detachment with merged replicas:



detachment *pervade* `Q` *clarify* *clarify* *pervade* `P` *occlude*

The merged blocks encourage pervasion of `Q` immediately. However they obscure the pervasion of `P`. Unlike `Q`, `P` is sitting upon separate blocks. The lower portion of `P` does not pervade the upper portion since the two do not share a common ground. The lower P and the block it sits upon, i.e. (P), does pervade the upper (P). This would require structure merging to be clearly and immediately visible. This example illustrates that specific dialects do have their own encumbrances that increase cognitive load. Dialects do require design as well as careful choice of applications. The need for design of iconic dialects is an analog of the need for selection of deductive steps during a symbolic demonstration. Some symbolic deductive problems are handled better by inferential techniques while others yield with less effort to resolution, and others still to Boolean transformations. The need to have a variety of symbolic tools strengthens the need also for a variety of iconic dialects.

There are many other imaginative unary dialects that can be designed. It can be both informative and entertaining to develop new dialects based on physical analogies.[23] Rock walls might implement deletion of bricks by the force of gravity. Fish swimming upstream, mountain climbing, books stacked on the shelf, gopher tunnels underground. All of these can be configured to represent boundary forms and thus logical deduction. Like Cuisenaire rods used to teach elementary numeric concepts in preschool, logic can be learned through physical activity. The difficulty is that the concepts taught by physical dialects are those of void-based unary logic rather than the more familiar string-based logic embedded in language.

14.3 Structural Complexity

Chapter 7 includes a metric for measuring the difficulty of unary reduction for most of the common logical tautologies. Chapter 8 applies this metric to axiomatic systems. The metric differentiates two major categories of complexity: those tautologies and systems that do not require *constructive application* of a unary axiom to demonstrate their validity (levels 1 and 2) and those that do (level 3).[24] Only two tautologies, Biconditional and

Distribution require level 3 construction techniques. The Biconditional is a simplified case of Pivot.

The previous section demonstrates reduction in several different dialects using the level 1 tautology Detachment as a standard exemplar. In general these dialects are relatively simple to use for all but the level 3 tautologies. The dialects and techniques in this chapter then apply to all common tautologies and most axiom systems developed from a foundation of implication. This makes them appropriate content in introductory logic courses, but severely limits their use for general problem solving.

Logic was developed by its founders to be relatively simple, with axiom systems at level 1 and 2 in complexity. More generally human thought and logical deduction is not necessarily complex. This same observation was made during the development of AI expert systems. Professionals do not engage in complex nested inference.

Which dialects are likely to be more useful than others when logic problems are more challenging? Certainly the textual form of relational ordered pairs, the enclosure form developed by Peirce, and the parens form used by some folks who study Spencer-Brown's work qualify. Dnets as well have been rigorously demonstrated on very challenging problems, although that work is not well known. From my experience, the participatory forms of maps, rooms and paths are more difficult to visualize, although that observation is based on little evidence and even less experience. Block forms may fall between experiential and enclosure techniques and again evidence is lacking.

There is a thin line between easy and challenging logic. The unary axioms are easy in their simple form but generalizing their power with illusory forms requires practice to learn to see intuitively. The dialects in Figure 14-1 show the `XOR` biconditional as the maximal example of a Boolean operator. The clearest step into complexity is the Huntington/Robbins axiom, what is called Cancellation in Chapter 5. Figure 13-7 shows the forms of Cancellation, Arrangement and Pivot for the textual notations.

There is an even simpler level 3 example, the Absorption theorem that has been used throughout the volume to exemplify the simplest non-trivial demonstration. This section shows the demonstration of Absorption in each of the eight representative dialects. We'll be looking for inherent difficulties that show up in any dialect.

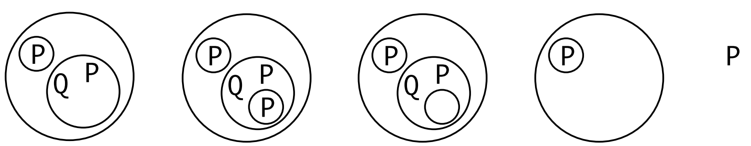
Here is the query demonstration of Absorption in Chapter 9.

```
((A) (      A B))
((A) (~(A)~ A B))                 query
((A) (~( )~ A B))                 pervade A
((A)           )                  occlude
  A                               clarify
```

Every query involves a constructive use of Pervasion, so it would require at least the ability to visualize construction of form within a dialect. I'll use reified Pervasion in demonstrations here because of its greater generality. The goal though is to develop an intuition about which dialects are easier to use in the constructive direction.

Incidentally I'm using `P` and `Q` as variables in this section. The variables `A` and `B` are unary pattern-variables. For some of the dialects it is unclear how easy it may be to substitute any pattern in place of a dialect variable.

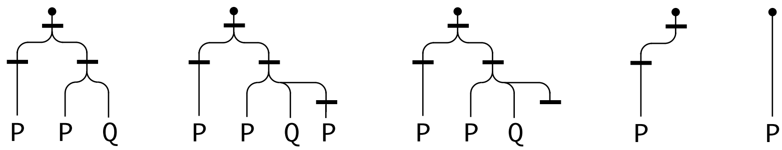
Enclosures



absorption *reify* `(P)` *pervade* `P` *occlude* *clarify*

Construction within the enclosure dialect is very easy, as would be expected for Peirce's choice of notation.

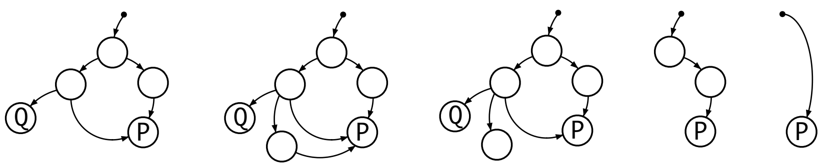
Capform Trees



absorption *reify* (P) *pervade* P *occlude* *clarify*

The tree representation replicates variable labels. This avoids the difficulties of added construction but does not qualify as a boundary form. Replication undermines structure merging. Managing multiple references is an extra burden on an implementation, and the physicality constraint discourages replicated objects. However it is very easy to add constructive structure to trees.

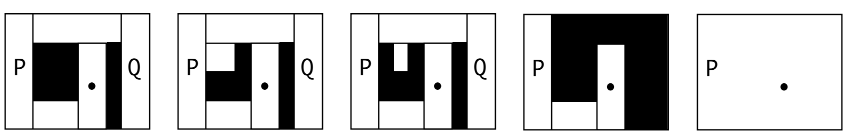
Distinction Networks



absorption *reify* (P) *pervade* P *occlude* *clarify*

Dnets have been a focus in this chapter. They are relatively easy to add structure to by adding links. With path following to determine pervaded nodes, reduction can be asynchronous. The axioms are easy to locate and apply in dnets, however adding structure is better done prior to merging.

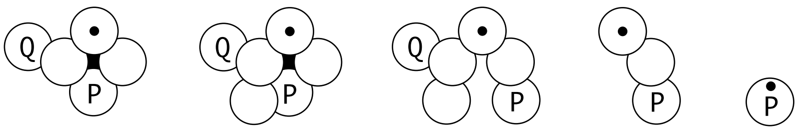
Planar Centered Maps



absorption *reify* (P) *pervade* P *occlude* *clarify*

It is difficult to layout a boundary form in the centered map dialect because of the added complexity of non-territorial black space to keep active territories separated. It also takes practice to understand the effect of reductions on the map, particularly since changes impact black space as well as territories. Planar maps as a dialect have little to recommend them.

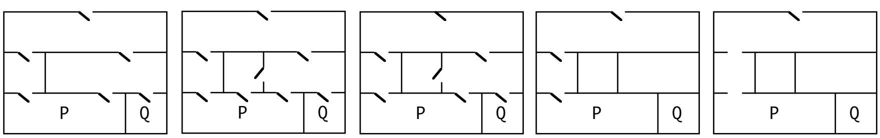
Stepping Stones



absorption *reify* (P) *pervade* P *occlude* *clarify*

The stepping version of maps does help to keep directionality clear. Some layouts create distortions of the territories so they no longer appear to be steps. The dialect is remarkably compact, and Pervasion is particularly easy to embody in the changing layout, both for reduction and for reification. However the overlapping steps do not support more complicated forms.

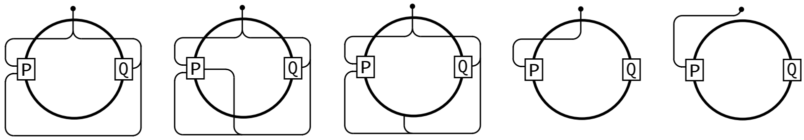
Doors and Rooms



absorption *reify* (P) *pervade* P *occlude* *clarify*

Directional doors resolve some of the difficulties with planar maps. Compartmentalization of rooms with walls that are impenetrable addresses the issue of black space in maps. Understanding door rules that implement axiomatic change is unfamiliar and requires practice. It is tempting to implement changes by removing rooms rather than by closing doors. Empty and closed off rooms just do not feel right. I suspect the immersive version would be very challenging.

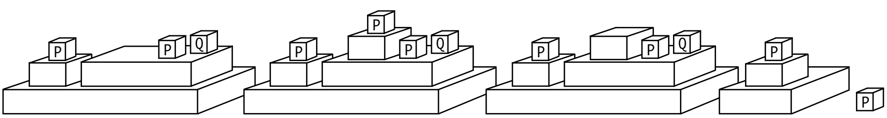
Capform Paths



absorption   *reify* (P)   *pervade* P   *occlude*   *clarify*

The inversion of thought that comes from merging distinctions within capform paths is refreshing and leads to additional insight about how unary logic works. Construction is very easy. Reduction is also relatively easy although it requires a global perspective. Like rooms, following paths to implement rules seems to be challenging. The immersive dialects do not seem to work very well when a person is expected to travel through rooms or along paths because the transformations require memory of the journey. Immersion in distinction networks does work because decision-making for any node is based purely on local connectivity.

Stacked Blocks



absorption   *reify* (P)   *pervade* P   *occlude clarify*

Stacked blocks are pleasing to use and work well interactively in their quasi-iconic dialect. Blocks are basically manipulable trees. Construction and reduction are both easy. Structure merging is easy to see but not necessarily easy to build.

Other than dnets, I have the most experience with physically stacked blocks. I've used books, kid's toy blocks, large legos, cardboard boxes, and literally whatever was in reach and stackable.

Summary

This summary is entirely subjective, I'm sharing my very sporadic exploration of these dialects over the last two decades. For understanding and working with boundary forms, I've tried to eliminate as much of the incidental structure of textual notation from logic as is possible. A lot of earlier work focused on just separating our historic symbolic forms of communication from what Frege and Peirce and Venn and earlier Euler identified as a different perspective. Most of the removal of linear incidental structure was done in conjunction with designing computational languages with the same goal, to support iconic programming and massive parallelism. Scott Kim wrote his innovative 1988 computer science thesis at Stanford based on the idea that the interface to computation should graphically resemble what the computer was doing rather than today's aging WIMP interface that insulates the user from the computation.[25]

Enclosures, Dnets, Stacked Blocks

These are recommended. Enclosures have been used for over a century to represent boundary logic. I've been using dnets in hardware/software co-design for over three decades. The new kid on the block is stacked blocks, which seem to have very desirable cognitive and manipulative features. I've used block logic in technical presentations for over two decades.

Rooms, Paths

These are recommended for exploration with the caveat that they may introduce difficulties in use and understanding. Rooms are evolved maps so they have some refinement at least from the global perspective. I'm not optimistic about their feasibility for maze-like exploration from the inside. Paths are quite innovative and are well worth further exploration, both globally and experientially.

Trees, Planar Maps, Projected Maps
I can't recommend these dialects. Trees are quasi-iconic and do not qualify as interesting. Planar maps hide potentially ambiguous configurations and do not necessarily enhance visualization. Projected maps do have interesting features that may be worth exploring however they also have drawbacks both for displaying forms and for reducing forms.

14.4 Remarks

The central idea for the chapter is that there is a rich world of iconic dialects to be explored, developed and invented. Logic, and other mathematical systems, can be expressed in non-symbolic ways that remain formal. Louis Kauffman reminds us: "... mathematics can behave non-trivially under change of notation."[26] These new tools may encourage different perspectives for rational thinking and cognition, and do provide a path for expressing formality as physical experience.

All of the examples in the volume are restricted to propositional logic. This does not imply that the iconic techniques are also restricted. Peirce and many others have extended Existential Graphs and other iconic approaches to quantification and relations in predicate logic.[27] My earlier work with parens forms and dnets was with optimization of the programming language LISP, including control logic, functions and relations. We used parens notation in another project to build a predicate calculus theorem prover.

Endnotes

1. **opening quote:** L. Kauffman (1997) The Robbins Problem - Computer Proofs and Human Proofs. p.2. Online 6/24 https://homepages.math.uic.edu/~kauffman/Robbins.htm

2. **words that do have referents:** W. Kneale & M. Kneale (1962) *The Development of Logic* p.233.

3. **structure of the syllogisms:** For a thorough academic review: A. Moktefi & S.J. Shin (2012) A history of logic diagrams p.611-682 of *Handbook of the History of Logic Volume 11.* Online 9/25 https://www.academia.edu/11636261/A_History_of_Logic_Diagrams For a more conversational presentation: Martin Gardner (1958) *Logic Machines and Diagrams*, online 9/25 https://archive.org/details/logicmachinesdia00mart For active research: *The Logical Geometry Project*, online 9/25 https://logicalgeometry.org/research/diagrams

Square of Opposition: Gregor Reisch (1503) *Margarita Philosphica* p.083. In Europe this book is one of the first encyclopedias of knowledge and was widely used as a university primer for several hundred years. Online 9/15 https://www.google.com/books/edition/Margarita_philosophica_nova/otxkBDdDZdUC?kptab=editions&gbpv=1

Geometry of Logic: Alonzo de la Veracruz (1569) *Recognitio Summularum* p.36. The image shows the relationship between the syllogisms. The book is about Physics and the image is common for the time.

Donkey Bridge: Louvain-la-Neuve (c.1500) Pons Asinorum Collection Magister Dixit Project, Ms. C24 *Dialectica* fol.131. *Dialectica* is a collection of student notes on logic lectures at the University of Louvain from the 15th to the 18th centuries. The image shows an art student who, like a donkey fearful of crossing a bridge that it cannot see the other side of, is likely to fall into the River of Vices when confronted by the geometry of the syllogisms.

Most sources for diagrammatic logic, including this text, begin with modern logic diagrams from Euler c.1750 through Frege and Peirce, c.1900. This is quite reasonable because modern logic was not formulated until the late nineteenth century. Prior to Euler, logic consisted almost entirely of Aristotle's Organon c.320 BCE and the study of categories and syllogisms. Another communality of books that specifically address logic diagrams is the absence of explanatory diagrams. 10,000 words is worthy of a picture.

4. **express the same logical intentions:** W. Bricken (1992) Spatial representation of elementary algebra. *1992 IEEE Workshop on Visual Languages* p.56-62. J. James & W. Bricken (1992) A boundary notation for visual mathematics. *1992 IEEE Workshop on Visual Languages* p.267-269.

5. **display of change over time:** The recorded video lectures mentioned in Chapter 1.2 include several animations of the dynamics of the unary axioms.

6. **maps, flows, paths and blocks:** W. Bricken (2006) Syntactic variety in boundary logic. In D. Barker-Plummer et al (eds) *Diagrams 2006*. LNAI 4045 p.73-87. Several of these dialects were motivated by the work of, and conversations with, Louis Kauffman.

7. **form of `NOR (A B)`:** `XOR` is the `NOR` of `NOR` and `AND`, and the `AND` of `OR` and `NAND`.

8. **that connect pairs of nodes:** Louis Kauffman originally considered forms as networks in L. Kauffman (1978) Network synthesis and Varela's calculus. *International Journal of General Systems* 1978 vol 4 p.179-187.

9. **by merging replicas of variables:** The source of replica labels in logic is `IFF` and its negation `XOR`. The symbolic logic notation for `X = Y` requires two comparisons, X→Y and Y→X, not separated by time. The indiscretion of multiple reference by multiple labels is the source of necessarily abstract logic (i.e. we cannot replicate physical objects, only their labels) and the ultimate source of computational complexity in logic problems.

10. **implement the axiom of Void Calling:** It does not make pragmatic sense to merge marks since Occlusion is much more powerful.

11. **and memory management:** How we think logically depends upon what we think logic is. Computation is not implication nor is it textual. From this perspective how we treat memory and memory storage tasks is a *thinking skill*. This in turn depends on our cognitive support devices, specifically making a list of labels on paper versus delegating reference to non-cognitive compiler processes in silicon. There is a long history in computer science about how we should treat computer memory. Back before we had terabyte thumb drives, references had to be counted within software programs. Now our phones provide that function.

12. **massively parallel processor:** Dnets were intended for very-fine-grain parallelism supported by tens of thousands of very weak processors, very much like a cellular automata. Back in the dark ages of parallelism 16 coarse-grain nodes was all we could get for physically testing asynchronous reduction. Coordination between these processors was the open problem since gathering results overwhelmed the advantage gained by parallelism. Dnets require no coordination and no global executive. W. Bricken & E. Gullichsen (1989) An introduction to boundary logic with the Losp deductive engine. *Future Computing Systems* 2(4), 1-77.

13. **reduced arithmetic form, a single mark:** A more detailed description of parallel algebraic reduction is intended for Volume II.

14. ***optimization achieved by local deletions:*** W. Bricken (1995) Distinction networks. In I. Wachsmuth, C.R. Rollinger & W. Brauer (eds) *KI-95: Advances in Artificial Intelligence* p.35-48.

15. **exploration of the process of deduction:** W. Winn & W. Bricken (1992) Designing virtual worlds for use in mathematics education: the example of experiential algebra. *Educational Technology* **32**(12) p.12-19.

16. **displayed for Peirce's circle dialect:** Peirce developed two dual notations, first Entitative Graphs with a disjunctive sheet of assertion, and later Existential Graphs with a conjunctive sheet of assertion. One is the unary Pivot of the other. Both are interpretations of containment forms. Here we are consistently using the disjunctive interpretation when reading forms as conventional logic; `(a b)` is read as `NOR a b`.

17. **links represent a** `WIRE-OR`**:** When two wires carrying a current are crossed, current will pass through when either wire is hot, which implements the logical `OR` of the wires. This of course is not a good idea for silicon architecture since wire-`OR` bypasses timing and power management. It is however the dominant way that home wiring passes signals through light switches.

18. **from a Boolean look-up table:** Transistor networks have a greater similarity to unary logic than to symbolic logic. Current either exists or does not exist on the wires that change the state of a transistor. The current changes the transistor from an insulator to a conductor. In addition silicon circuits have equally complex networks that manage timing, power distribution and cross chip communication. To understand deduction and rationality we need to explicitly include its infrastructure.

19. **formalization of predicate calculus:** Frege's original notation flowed from right to left. I changed it to read from left to right. His crossbars intersected but did not cross the flow line they intersect. I centered the bar across the flow line. Finally I merged the multiple references to inputs and added a heavier stroke to the crossbar.

20. **rapid termination of the search:** I can't imagine this to be a good game unless you knew that there was a clear solution available. We would need a rule for structure merging for a complete strategy, as well as the option of constructing doors and rooms.

21. **sides for the enclosure family:** L. Kauffman (unpublished) *Laws of Form - An Exploration in Mathematics and Foundations Section VII.* Online 3/25 at `http://homepages.math.uic.edu/~kauffman/Laws.pdf`

22. **drawn through all capform bars:** L. Kauffman (1993) *Knots and Physics* 2nd edition p.605ff. The parens representation is first over-capped to convert each parens into a single bar. Then, rather than under-capping matching bar ends to construct the circle variety, under-caps are constructed iteratively from the deepest space to connect adjacent parens fragments: connecting `)(` in the case of `()()`, and `))` in the case of `(())`. This regime maps even depths to the outside of the curve and odd depths to the inside, with nesting depth counted by the number of transverse crossings of the universal distinction.

23. **based on physical analogies:** Over the years I've explored viable dialects based on rock walls under gravity, flowing rivers, fish swimming up rivers, waterfalls, mountain ranges, stacks of books and fruit, several innovative varieties of functional silicon circuitry, cellular automata, mazes, virtual reality environments, board games, cryptographic puzzles, and many programming languages.

24. **and those that do (level 3):** I'm using the word complexity in the context of unary reduction of propositional forms and not the more widely known mathematical complexity theory that characterizes the difference between polynomial and exponential effort of algorithms. There is however a communality in that SAT problems are the standard for minimal mathematical complexity, and these problems are within the same domain of propositional logic that is addressed by unary reduction in this volume.

25. **insulating the user from the computation:** WIMP is Windows, Icons, Menus and Pointers. Fifty years later we are still using those tools even though each year computation evolves the equivalent of at least ten human years. Scott Kim's 1988 thesis is *Viewpoint: Toward a Computer for Visual Thinkers*. Online 9/25 `https://worrydream.com/refs/Kim_1988_-_Viewpoint,_Toward_a_Computer_for_Visual_Thinkers.pdf` Video demo of Viewpoint: `http://youtu.be/9G0r7jL3xl8`

26. **under change of notation:** L. Kauffman (1997) The Robbins Problem - Computer Proofs and Human Proofs. Online 6/24 `https://homepages.math.uic.edu/~kauffman/Robbins.htm`

27. **relations in predicate logic:** P. Kramer (2022) Iconic mathematics: math designed to suit the mind. *Frontiers in Psychology* 6/13 13:890362

Bibliography

All entries in the bibliography are from the chapter endnotes.

G. Allwein & J. Barwise (eds) *Logical Reasoning with Diagrams.*

A. H. Almaas (1986) *The Void: Inner spaciousness and ego structure*.

Aristotle (c.350 BCE) *Physics Book IV.* R. Hardie & R. Gaye (trans).

_______ (c.350 BCE) *Metaphysics*. W. D. Ross (trans).

Augustine (389 CE) *De Magistro (The Teacher)*. Quote from B. Rabern (2022) Pure quotation in linguistic context. *Journal of Philosophical Logic.*

B. Banaschewski (1977) On G. Spencer-Brown's Laws of Form, *Notre Dame Journal of Formal Logic* 18:3.

J. Barbour (2011) Bit from it. `http://platonia.com/bit_from_it.pdf`

J. Barrow (2000) *The Book of Nothing.*

J. Barwise & J. Etchemendy (1994) *Hyperproof.*

_______ (1995) Heterogeneous logic. In B. Chandrasekaran *et al* (eds) *Diagrammatic Reasoning.*

_______ (1996) Visual information and valid reasoning. In G. Allwein & J. Barwise (eds) *Logical Reasoning with Diagrams.*

G. Bateson (1970) Form, substance, and difference. *In Steps to an Ecology of Mind* (1972).

_______ (1972) Redundancy and coding, in *Steps to an Ecology of Mind*.

M. Beeson, R. Veroff & L. Wos (2005) Double-negation elimination in some propositional logics. `https://arxiv.org/abs/cs/0301026`

G. Boole (1854/1958) *The Laws of Thought.*

N. Bourbaki (1950) The Architecture of Mathematics. *American Math.Monthly* 57.

P. Bouvier & H. Garavel (2021) The VLSAT-2 Benchmark Suite. Tech.Report 0514. `https://cadp.inria.fr/ftp/publications/others/Bouvier-Garavel-21-c.pdf`

W. Bricken (1992) Spatial representation of elementary algebra. *1992 IEEE Workshop on Visual Languages*.

_______ (1995) Distinction networks. In I. Wachsmuth, C.R. Rollinger & W. Brauer (eds) *KI-95: Advances in Artificial Intelligence*

_______ (2006) Syntactic variety in boundary logic. In D. Barker-Plummer et al (eds) *Diagrams 2006.* LNAI 4045.

_______ (2008) Simplicity rather than knowledge. *AI Magazine* **29**(2).

_______ (2017) Distinction is sufficient: Iconic and symbolic perspective on Laws of Form. *Cybernetics and Human Knowing* 24:3-4.

_______ (2019, 2021) *Iconic Arithmetic Vols I, II, III.*

_______ (2023) The Use of Boundary Logic. *Laws of Form: Fiftieth Anniversary.*

_______ (2025) *Laws of Form Conference 2024*
Unary Boundary Logic-I: Void-based Axioms https://vimeo.com/1040314440
Unary Boundary Logic-II: Illusory Forms https://vimeo.com/1040314031
Unary Boundary Logic-III: Virtual Queries https://vimeo.com/1039534775

W. Bricken & G. Coco (1995) VEOS: The virtual environment operating shell. In W. Barfield & T. Furness (ed) *Virtual Environments and Advanced Interface Design.*

W. Bricken & E. Gullichsen (1989) An introduction to boundary logic with the Losp Deductive Engine. *Future Computing Systems* 2(4).

F. M. Brown (1986) An experimental logic based on the fundamental deduction principle. *Artificial Intelligenc* 30.

_______ (1990) Boolean reasoning. *The Logic of Boolean Equations.*

G. Burnett-Stuart (2020) Simplification algorithm for CL-expressions. http://www.markability.net/pervade

_______ (2023) Pervasive Simplicity. *Laws of Form A Fiftieth Anniversary*.

_______ (2024) Five liars. http://www.markability.net/five_liars.htm

B. Caplan (2007) *The Myth of the Rational Voter.*

I. Calvino (1965) A sign in space. *Cosmicomics.* W. Weaver (trans).

R. Carnap (1937) *The Logical Syntax of Language.*

L. Carroll (1896) *Symbolic Logic* Part II Advanced, Book XIII.

B. Chandrasekaran, J. Glasgow, & N. Narayanan, (eds) *Diagrammatic Reasoning.*

G. Chaucer (c.1378) Tale of Melibee. *The Canterbury Tales.*

F. Close (2009) *Nothing.*

E. F. Codd (1970) A relational model of data for large shared data banks. *Communications of the ACM* 13(6).

K.C. Cole (2001) *The Hole in the Universe.*

R. Corazzon *History of Logic.* https://www.historyoflogic.com/

L. Couturat (1905/1914) *The Algebra of Logic*. L. Robinson (trans).

L. DaVinci *The Notebook*. (1478-1510/1954) E. Macurdy (ed, trans).

M. Davis, R. Sigal, & E. Weyuker (1994) *Computability, Complexity, and Languages* 2nd edition.

P. Davis (1997) Mathematics in an age of illiteracy. *SIAM News* 30(9) 11/97.

_______ (1998) A brief look at mathematics and theology. Addresses given at Scripps College, Spring 1998.

A. de la Veracruz (1569) Geometry of Logic. *Recognitio Summularum.*

K. Devlin (2006) The useful and reliable illusion of reality in mathematics. *Toward a New Epistemology of Mathematics Workshop, GAP6 Conference* 9/2006.

M. Domaratzki & A. Okholin (2004) Representing recursively denumerable languages by iterated deletion. *Theoretical Computer Science archive* 314(3).

P. Duhem (1909-1916/1985) R. Ariew (ed, trans) *Medieval Cosmology.*

G. Ellsbury (1983) The primary algebra demonstrated from a single initial. Privately distributed in 1986.

Encyclopedia Britannica (1771) Volume II, Logic.

_______ https://www.britannica.com/topic/history-of-logic

J. Engstrom (1999) G. Spencer-Brown's Laws of Form as a revolutionary, unifying notation. *Semiotica* 125(1/3).

A. Eschen & J. Villadsen (2022) On axiomatic systems for classical propositional logic. *Proceedings of Isabelle Workshop 2022*.

T. Etter (2006) Three-place identity. https://web.archive.org/web/20130510223843/http://www.boundaryinstitute.org/bi/articles/Three-place_Identity.pdf

L. Euler (1802) Letter CIV Different forms of syllogisms (2/21/1761). H. *Letters of Euler.* Hunter (trans).

B. Fitelson (1997) Using Mathematica to Understand the Computer Proof of the Robbins Conjecture. https://fitelson.org/robbins.pdf

_______ (1997) The Robbins Conjecture -- A Brief History. https://fitelson.org/devconf.pdf

_______ (2003) Some recent results in algebra and logical calculi obtained using automated reasoning. Presentation at U.C. Berkeley. https://fitelson.org/presentation.html

G. Frege (1879) *Concept-Script: A Formal Language for Pure Thought Modeled on that of Arithmetic.*

_______ (1884) *The Concept of Number.*

_______ (1884/1950) *The Foundations of Arithmetic.*

_______ (1903) *Basic Laws of Arithmetic Vol.II.* P.A. Ebert & M. Rossberg (trans).

A. From, A. Moesgård & J. Villadsen (2021) Formalizing axiomatic systems for propositional logic in Isabelle/HOL. *Intelligent Computer Mathematics.*

D. M. Gabbay and J. Woods (2004–2012) *Handbook of the History of Logic.*

M. Gardner (1958) *Logic Machines and Diagrams.*

H. Genz (1999) *Nothingness.* K. Heusch (trans).

J. Goguen (c.2001) On Notation. https://cseweb.ucsd.edu/~goguen/projs/uid.html

W. E. Gould (6/ 1977) Review of Laws of Form. *Journal of Symbolic Logic* 42:2.

M. Greaves (2002) *The Philosophical Status of Diagrams.*

K. Hara (2009) *White.*

Y. Harari (2015) *Sapiens: A brief history of humankind.*

G. Harman (2018) *Object-oriented Ontology.*

D. Hilbert & W. Ackermann (1928/1999) *Principles of Mathematical Logic.*

E. Huntington (1933) New sets of independent postulates for the algebra of logic, with special reference to Whitehead and Russell's Principia Mathematica. *Transactions of AMS* 35.

_______ (1933) Boolean algebra. A correction. *Transactions of AMS* 35.

Y. Imal & K. Iséki (1965) On axiom systems of propositional calculi I. *Proceedings of the Japan Academy* 41/6.

Internet Encyclopedia of Philosophy. https://iep.utm.edu/propositional-logic-sentential-logic/

J. James & W. Bricken (1992) A boundary notation for visual mathematics. *1992 IEEE Workshop on Visual Languages*.

M. Johnson (1987) The body in the mind. In F. Varela, E. Thompson & E. Rosch (1991) *The Embodied Mind.*

D. Kahneman (2011) *Thinking, Fast and Slow.*

R. Kaplan (1999) *The Nothing That Is: A natural history of zero.*

L. Kauffman (1978) Network synthesis and Varela's calculus. *International Journal of General Systems* 1978 vol 4.

_______ (1986) *Map Reformulation.* Princelet Editions #30.

_______ (1987) On Knots W. Browder, R. Langlands, J. Milnor & E. Stein (eds) *Annals of Mathematics Studies* #115.

_______ (1987) Self-reference and recursive forms. *J. Soc. Biol. Structure*s 10.

_______ (1988) Extension is a single initial for the primary algebra. Unpublished. https://homepages.math.uic.edu/~kauffman/Extension.pdf

_______ (1990). Robbins algebra. *Proceedings of the Twentieth International Symposium on Multiple Valued Logic*.

_______ (c.1992) The Robbins Problem -- Computer Proofs and Human Proofs. https://homepages.math.uic.edu/~kauffman/Robbins.htm

_______ (1993) *Knots and Physics* 2nd edition.

_______ (1995) Arithmetic in the form. *Cybernetics and Systems* 26(1).

_______ (1997) The Robbins Problem - Computer Proofs and Human Proofs. https://homepages.math.uic.edu/~kauffman/Robbins.htm

_______ (2001) The mathematics of Charles Sanders Peirce. *Cybernetics & Human Knowing* 8, no.1–2.

_______ (2002) Biologic. arXiv:quant-ph/0204007v2

_______ (2002) Time, imaginary value, paradox, sign and space. *American Institute of Physics Conference Proceedings* 627(1).

_______ (2003) Reformulating the Map Color Theorem. https://arxiv.org/abs/math/0112266

_______ (2024) The myth of the boundary -- Laws of Form, logic and sets. *Distinction Journal of For*m 1(1) Spring 2025.

_______ (unpublished) Laws of Form -- An Exploration in Mathematics and Foundations Section VII. http://homepages.math.uic.edu/~kauffman/Laws.pdf

L. Kauffman & F. Varela (1980) Form dynamics. *J. Social Biol. Struct*. 3.

S. Kim (1988) *Viewpoint: Toward a Computer for Visual Thinkers.* https://worrydream.com/refs/Kim_1988_-_Viewpoint,_Toward_a_Computer_for_Visual_Thinkers.pdf

S.C. Kleene (1952) *Introduction to Metamathematics.*

I.Kleiner (1991) Rigor and proof in mathematics: A historical perspective. *Mathematics Magazine* 64.

W. Kneale & M. Kneale (1962) *The Development of Logic.*

H. Kragh (2013) RePoSS#21; Empty space or ethereal plenum? Early ideas from Aristotle to Einstein.

P. Kramer (2022). Iconic mathematics: math designed to suit the mind. *Frontiers in Psychology* 6/13 13:890362.

R. L. Kuhn (2013) Levels of Nothing. *Skeptic Magazine* 18(2) 9/2013.

I. Lakatos (1976) *Proofs and Refutations: The logic of mathematical discovery.*

G. Lakoff & R. Núñez (2000) *Where Mathematics Comes From.*

G. Lakoff & M. Johnson (2003) *Metaphors We Live By.*

J. Lanier (2010) *You are Not a Gadget.*

J. Larkin & H. Simon (1987) Why a diagram is (sometimes) worth ten thousand words. In Chandrasekaran et al *Diagrammatic Reasoning.*

G. Leibniz (1679) Correspondence to Tschirnhaus (Math., IV, 481; Brief., I, 405), as quoted in L. Couturat (1901) *The Logic of Leibniz.*

_______ (1903) De formæ logicæ comprobatione per linearum ductus (Phil., VII, B, IV, 1-10). In L. Couturat (1903) *Opuscules et fragments Inédits de Leibniz.*

T. Lenoir (2008) Machinic bodies, ghosts, and paraselves: confronting the singularity with Brian Rotman. Foreword to B. Rotman (2008) *Becoming Beside Ourselves.*

W. Lenzen (2006) *Leibniz' Logic.*

J. Leslie & L.W. Kuhn (2013) (eds) *The Mystery of Existence: Why is there anything at all?*

R. Lindsay (1995) Images and inference. In B. Chandrasekaran *et al* (eds) (1995) *Diagrammatic Reasoning.*

X. Liu (2017) An axiomatic system for Peirce's alpha graphs. https://www.researchgate.net/profile/Xinwen-Liu/publication/250714372_An_Axiomatic_System_for_Peirce's_Alpha_Graphs/links/59d7a4d2458515bbfee8ddc2/An-Axiomatic-System-for-Peirces-Alpha-Graphs.pdf

R. Llull (1501) *Ars magna.* Barch[ino]ne: Impressum per Petru[m] Posa.

P. Lockhart (2012) Measurement.

Logical Geometry Project. https://logicalgeometry.org/research/diagrams

J. R. Lucas (2000) *The Conceptual Roots of Mathematics.*

O. Makridis The Sheffer stroke. *Internet Encyclopedia of Philosophy.* https://iep.utm.edu/sheffers/

Mathematica documentation. https://reference.wolfram.com/language/guide/TheoremProving.html

H. Matisse. https://www.thehistoryofart.org/henri-matisse/quotes

H. Matsuo (1987) *The Logic of Unity.* K. Inada (trans).

H. Maturana & F. Varela (1987) *The Tree of Knowledge*.

B. Mazur (2003) *Imagining Numbers.*

W. McCune (1997) Solution of the Robbins problem. *Journal of Automated Reasoning* 19(3).

P. McDonnell (2005) *The Gift of Nothing.*

M. McLuhan (1967) *The Medium is the Massage*

Metamath documentation. https://us.metamath.org/index.html

J. Mingers & C. Standing (2018) Framework for validating information systems research based on a pluralistic account of truth and correctness. *Journal of the Association for Information Systems* 19(4)

C. Misner, K. Thorne & J. Wheeler (1970) *Gravitation.*

A. Moktefi & S.J. Shin (2012) A history of logic diagrams. In *Handbook of the History of Logic* Volume 11. https://www.academia.edu/11636261/A_History_of_Logic_Diagrams

T. Morton (2013) *Hyperobjects*

Muktananda Paramahasna (1975) *Introduction to Kashmir Shaivism.*

T. Needham (1999) *Visual Complex Analysis.*

A. Newell & H. Simon (1972) *Human Problem Solving.*

J. Nicod (1916-17) *A Reduction in the Number of the Primitive Propositions of Logic.* Trinity College.

T. Norretranders (1991) *The User Illusion.* J. Sydenham (trans).

T. Öner (2018) Independent sets of axioms for boolean algebras. https://dergipark.org.tr/tr/download/article-file/489393

Online Encyclopedia of Integer Sequences. A000081, A000108 https://oeis.org/

G. Păun (2006) Introduction to membrane computing. In G. Ciobanu, G. Păun & M. Pérez-Jiménez (eds) *Applications of Membrane Computing* (2006).

C. S. Peirce (1904) The new elements of mathematics. In C. Eisele (ed) (1976) *Mathematical Philosophy 4.*

_______ (1958) *The Collected Papers of Charles Sanders Peirce.* Electronic edition reproducing Vols. I-VI (1931-1935) C. Hartshorne & P. Weiss (eds) , Vols. VII-VIII (1958) A. W. Burks (ed).

F. J. Pelletier (1990) *Parmenides, Plato, and the Semantics of Not-Being.*

_______ (1999) A brief history of natural deduction. History and *Philosophy of Logic* vol.20.

A-V Pietarinen (2020) Getting closer to iconic logic. *Diagrammatic Representation and Inference. Diagrams 2020. Lecture Notes in Computer Science,* vol 12169.

F. Pfenning (1988) Single axioms in the implicational propositional calculus. *9th International Conference on Automated Deduction* 1988.

Pons Asinorum Collection Magister Dixit Project. Donkey Bridge: Louvain-la-Neuve (c1500) Ms. C24 *Dialectica* fol.131.

S. Powell (2012) Equations and the equal sign in elementary mathematics textbooks. *Elementary School Journal* 112(4).

Prajñāpāramitāhṛdaya Sutra of Mahāyāna Buddhism as told by the 14th Dalai Lama, Tenzin Gyatso. Geshe Thupten Jinpa (trans). https://www.lamrim.com/hhdl/heartsutra.html

A. Prior (2014) Peirce's axioms for propositional calculus. *Journal of Symbolic Logic* 25/2.

W. V. Quine (1960) *Word and Object.*

_______ (1982) *Methods of Logic* Fourth Edition.

Gregor Reisch (1503) Square of Opposition. *Margarita Philosphica* p.083. https://www.google.com/books/edition/Margarita_philosophica_nova/otxkBDdDZdUC?kptab=editions&gbpv=1

E. Richards (1930) *Eletelephony* (poem).

J. Robinson (1979) *Logic, Form and Function.*

B. Rotman (1987) *Signifying Nothing.*

_______ (2000) *Mathematics as Sign*

R. Rucker (1987) *Mind Tools.*

D. G. Schwartz (1981) Isomorphisms of Spencer-Brown's Laws of Form and Varela's Calculus of Self-Reference. *Int. Journal of General Systems* 6.

C. Seife (2000) *Zero*.

B. Selman, H. Kautz & D. McAllester (1997) Ten challenges in propositional reasoning and search. *Proceedings of IJCAI-97*.

S. J. Shin (2002) *The Iconic Logic of Peirce's Graphs.*

_______ (2022) Peirce's deductive logic, Section 2, *Stanford Encyclopedia of Philosophy.* (Summer 2022 Edition). E. N. Zalta (ed)

S. J. Shin, O. Lemon & J. Mumma (2008) Diagrams and diagrammatical reasoning. *Stanford Encyclopedia of Philosophy* Fall 2025. E. Zalta & U. Nodelman (eds) https://plato.stanford.edu/archives/fall2025/entries/diagrams/

Jan Simner (c.2022) *The Sociological Interpretation of George Spencer-Brown's Calculus of Indictions*.

H. Simon (1995) Forward to B. Chandrasekaran *et al* (eds) (1995) *Diagrammatic Reasoning.*

T. Skolem (1928) On mathematical logic. In J. van Heijenoort (ed) (1967) *From Frege to Gödel.*

J. Smith (2019) *Irrationality.*

J. Sowa (2005) Existential Graphs. https://jfsowa.com/peirce/ms514.htm

G. Spencer-Brown (1969) *Laws of Form.* First Ed. London: Allen & Unwin.

Stanford Encyclopedia of Philosophy. https://plato.stanford.edu/ and https://plato.stanford.edu/entries/lukasiewicz/

J. Stillwell (2016) *Elements of Mathematics.*

P. Suppes (1957) *Introduction to Logic.*

A. Tarski, L. Henkin, & D. Monk (1971) *Cylindric Algebras.*

N. Tennant (1984) The withering away of formal semantics. *Mind and Language V.* 1(4) 1986.

F. Varela (1975) A calculus for self-reference. *Int. Journal of General Systems* 2.

_______ (1976) Not One, Not Two: position paper for the Mind-Body conference. In *The Coevolution Quarterly* Fall1976.

_______ (1992) *Ethical Know-How: Action, Wisdom, and Cognition.*

F. Varela, E. Thompson & E. Rosch (1991) *The Embodied Mind.*

S. Verlan (2010) Recent developments on insertion/deletion systems. *Computer Science Journal of Moldavia* 18 2(53) p.210-245.

E. W. Weisstein Spencer-Brown form. *MathWorld -- A Wolfram Web Resource*. https://mathworld.wolfram.com/Spencer-BrownFomh.html

H. Weyl (1941) *The Mathematical Way of Thinking.*

J. Wheeler (1990) Information, physics, quantum: the search for links. *Proc. of the Third Int. Symp. Foundations of Quantum Mechanics.*

A. N. Whitehead (1936) Harvard: The Future. *The Atlantic Monthly* Sept 1936

A.N. Whitehead & B. Russell (1910/1980) *Principia Mathematica.*

Wikipedia https://en.wikipedia.org/wiki/Sheffer_stroke

S. Winker (1992) Absorption and idempotency criteria for a problem in near-Boolean algebras. *Journal of Algebra* 153, 2.

W. Winn & W. Bricken (1992) Designing virtual worlds for use in mathematics education: the example of experiential algebra. *Educational Technology,* 32(12).

L. Wittgenstein (1922) *Tractatus Logico-Philosophicus.* C. Ogden (trans).

S. Wolfram (2002) *A New Kind of Science.*

_______ (2003) *The Mathematica Book* 5th edition.

_______ (2022) *Metamathematics: Foundations & Physicalization.*

_______ (2023) *Mathematica* v13.0.

R. Zach (1999) Completeness before Post: Bernays, Hilbert, and the development of propositional logic. *Bulletin of Symbolic Logic.* Sept. 1999.

Zhuang Zhou (c.300 BCE) Working everything out evenly. *The Book of Chuang Tzu.* M. Palmer (trans).

Organization of the Index

The **Primary Reference Figures** summarize the unary design principles, the axioms and theorems, and the transcription map between symbolic and unary representations of logic. The iconic dialects in Chapter 14 are also illustrated.

PRIMARY REFERENCE FIGURES

The **index** is organized into four major categories.
Each category is indexed by primary keywords.

PEOPLE with quotes in bold

SYMBOLIC LOGIC Concepts, historical context, operators, axiom systems.

COMPUTATION Unary logic was developed within a computational environment. Technical features like pattern-matching, substitution, efficiency, parallelism and automated theorem proving are not included within logic.

UNARY LOGIC Primary concepts and categories are organized by keywords.

AXIOMS AND THEOREMS Definitions, properties and behavior.

VOID-BASED REDUCTION Techniques, illusory forms, querying, void.

EXAMPLES The applications of unary logic in this volume.

NOTATIONS AND DIALECTS Iconic representations and reductions.

SYMBOLS AND ACRONYMS

If you can't find something, look under void.

==================================

PEOPLE

quotations in bold

bibliographic references not included

==================================

=================================

SYMBOLIC LOGIC

Figures in italics

=================================

COMPUTATION

UNARY LOGIC

==============================

AXIOMS and THEOREMS

==============================

==============================

VOID-BASED REDUCTION

==============================

EXAMPLES

================================

NOTATIONS and DIALECTS

================================

=============================

SYMBOLS and ACRONYMS

=============================

MATHEMATICS

LOGIC

UNARY

ICONIC

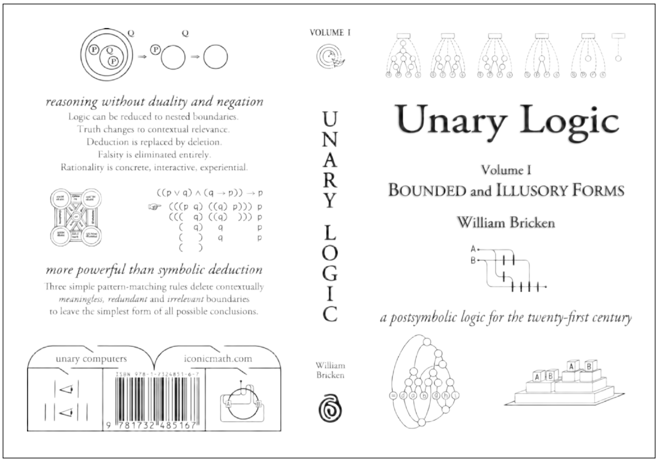
ACRONYMS

=============================

COVER

10 11 1

8 9



2

7 6 5 4 3

1. Ch.7 p.156: demonstration of Constructive Dilemma in the dnet dialect
2. Ch.14 p.375: representation of XOR in the crossbar network dialect, a reformatting of Frege's 1879 *concept writing* notation
3. Ch.14 p.379: representation of XOR in the stacked physical blocks dialect
4. Ch.11 p.272: dnet representation of a structural equivalence problem
5. Ch.3 p.56: Jordan curve icon
6. Ch.14 p.377: representation of XOR in the capform path dialect
7. Ch.3 p.53: inside or outside depends on perspective
8. Ch.14 p.353: *Square of Opposition* in Gregor Reisch (1503) *Margarita Philosphica*
9. Ch.7 p.157: demonstration of a variation of the Cases Theorem in parens notation
10. Ch.1 p.19: parallel demonstration of detachment using Peirce's circle notation
11. Ch.1 p.14: representation of the semipermability of Deep Pervasion in the circle dialect

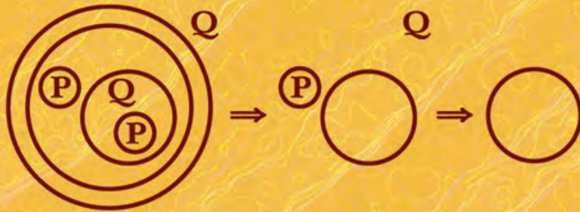
Back Cover Words

reasoning without dualism and negation

Logic can reduced to nested boundaries.
Truth changes to contextual relevance.
Deduction is replaced by deletion.
Falsity is eliminated entirely.
Rationality is concrete, interactive, experiential.

more powerful than symbolic deduction

Three simple pattern-matching rules delete contextually *meaningless*, *redundant* and *irrelevant* boundaries to leave the simplest form of all possible conclusions.



reasoning without duality and negation

Logic can be reduced to nested boundaries.
Truth changes to contextual relevance.
Deduction is replaced by deletion.
Falsity is eliminated entirely.
Rationality is concrete, interactive, experiential.

$$((p \vee q) \wedge (q \rightarrow p)) \rightarrow p$$

```
☞ (((p q) ((q) p))) p
  (((  q) ((q)  ))) p
    (  q)   q       p
    (   )   q       p
    (   )
```

more powerful than symbolic deduction

Three simple pattern-matching rules delete contextually *meaningless, redundant* and *irrelevant* boundaries to leave the simplest form of all possible conclusions.

unary computers

iconicmath.com

ISBN 978-1-7324851-6-7
9 781732 485167