LOSP FULL-DAY SEMINAR

William Bricken
November 1988

Presented at a commercial Uni-Ops seminar.

TECHNICAL SYNOPSIS

... a new analytic methodology, simpler yet more powerful than standard mathematical techniques, and its applications to symbolic logic, computer programming, parallel processing, and network design.

The seminar begins by introducing and demonstrating the fundamental concepts of the Laws of Form, which are the basis of the methodology. The first application area is propositional calculus. Techniques for expressing all Boolean expressions with a single operator are presented and applied to Boolean simplification and optimization. Logical deduction is performed entirely by canceling and transferring terms, using five rules that require no mathematical insight.

The methodology is extended to Predicate Calculus and applied to program optimization and theorem proving. Programs are converted from serial to parallel representations using connectionist-like networks. The single operator logic simplifies the design and complexity of conventional networks. Strong parallelism is demonstrated, in which local evaluation yields global consequences. Logic is then extended to higher orders by the introduction of imaginary Boolean variables that reenter their own indicative space. The paradoxical aspects of self-reference are resolved by the introduction of time into the representation of logic. Network design is extended to include recursive and infinite structures using the imaginary Boolean variable.

The content of the seminar goes underneath traditional mathematical formalisms by introducing the single concept of distinction and then constructing logic, set theory, and the arithmetic of numbers from this primal concept. The application of this new mathematical technique is discussed in the context of a broad range of disciplines.

Seminar participants will find that a working knowledge of propositional calculus and of programming techniques will be quite helpful. The only other prerequisites are a habit of rigorous thinking and a willingness to look underneath traditional mathematical concepts that are generally regarded as fundamental.

References

Spencer-Brown, Laws of Form, Dutton, 1972 (may be out of print)

Kauffman, Louis H. and F. J. Varela, Form Dynamics, Journal of Social and Biological Structures, V3 #2, April 1980, 171-206

Kauffman, Louis H., Network Synthesis and Varela's Calculus, International Journal of General Systems, 1978, V4, 179-187.

F. J. Varela, A Calculus for Self-reference, International Journal of General Systems, 1975, 2, 5-24.

Goguen, J.A. and F.J. Varela, Systems and Distinctions; Duality and Complementarity, International Journal of General Systems, 1979, 5, 31-43.

PROMOTIONAL DISTRIBUTION

Benefits of the Seminar

Laws of Form is a revolutionary new mathematics that models the evolution of symbolic thought. The seminar provides a thorough introduction to distinction and the mathematics of indication, from both a practical, intuitive perspective and a formal, computational perspective.

What You Will Learn

1. the conventions underlying symbolic thought

2. a new foundation for mathematics that is intuitive, constructive and holistic

3. the arithmetic, algebra, and calculus of indications

4. how to generate canonical and minimal forms for symbolic systems such as logic and programming languages, with the minimum of computational effort

5. fast, efficient deductive techniques

6. how to understand and calculate with paradoxical and self-referential forms

7. the use of imaginary Boolean values and oscillating truth values

8. how to think about thinking.

Who Should Attend

Programmers, mathematicians, philosophers, psychologists and teachers interested in innovative formal systems, deductive reasoning, and the foundations of programming and thought. Anyone who has thought about or read Spencer-Brown's Laws of Form. Some familiarity with Computer Science and symbolic logic is desirable.

Quotes

"Only difference makes a difference."
-- Gregory Bateson

"... a new calculus of great power and simplicity."
-- Bertrand Russell

"... that particular calculus which can let us see deeper into the nature of mathematics"
-- L. L. Whyte

"In Laws of Form, language, theories, observations and hypotheses are seen to arise naturally from the ground of our study which is ourselves."
-- Louis Kauffman

Descriptive Notes

Logic plays a fundamental role in computer science. Propositional calculus, the core of mathematical logic, is integral to the control of programs, the design of circuits, the reasoning of expert systems, and structure of search spaces. Formal logic is in its infancy, it is clumsy, overly complex, and computationally inefficient.

Laws of Form simplifies all calculations involving propositional calculus by eliminating the redundancy in logic. Based on the seminal work of G. Spencer-Brown, Losp is the implementation of Laws of Form; it is the assembly language of deduction. All logical structure arises out of its single concept of distinction.

The advantages of using Losp include the following:

1. Losp is fully general, complete and consistent. The single operator condenses the representation of propositional calculus.

2. Proof is achieved by a single uniform technique: the principled erasure of irrelevant information.

3. Since erasure is the primary form of simplifying transformation, the representation of a problem decreases in size (and memory is consequently released) during deductive computation.

4. The distinction between forward and backward chaining is irrelevant. All search is opportunistic.

5. The Losp representation maps directly onto circuits using n-ary nor gates, and onto networks using distinction nodes.

6. Expressions are partitioned for parallel processing by the distinction operator.

7. Paradoxical and self-referring expressions have a natural form of representation, in time rather than in space.

Losp has potential applications wherever propositional calculus and deduction are used. These applications include:

logic programming
theorem proving
Boolean minimization
program optimization
program verification
parallel processing
model based reasoning
machine learning
consistency maintenance

The axioms of Losp are constructive, intuitive, and extremely simple. Logic is only one interpretation for this fundamental mathematics; it can also be used for Boolean algebra, set theory, and number theory. Laws of Form has interpretations in psychology, education, metaphysics, systems theory, biology, and other disciplines.

The Losp seminar will provide fundamental tools for those interested in logic programming and Boolean algebra. Since Laws of Form starts at the very beginning of the concept of symbolizing, the rules and understanding exposed at this fundamental level will help to clarify the meaning and utility of symbol systems and mathematics. The concepts of Losp provide a unique vocabulary for thinking about thought, and the position of the thinker in relation to knowledge.

CONTENTS

This outline contains a list of topics to be covered in the seminar, organized into 1 1/2 hour modules.

III: The Mathematics of Indication, Part 2

- Algebra and evaluation
- Equality
- Axiomatic basis
  - dominion
  - pervasion
  - transposition
- Theorems
  - involution
  - reference
  - absorption
  - cancellation
  - flex
- Transposition and erasure
- Semantics
- Canonical form
- Proof techniques
  - Transformation
  - Implication
  - Case Analysis
- Boolean algebra
- Representational incompleteness
- Set syntax and arbitrary arity
- Boolean minimization

IV: Applications in First Order Logic

- Condensation of concepts
- Maps
- Tautology, contradiction and contingency
- Transformational proof
- Algebraic Inference
- Interface placement and access rules
- The Losp Deductive Engine
- Quantification
- Canonical forms
- Theorem proving
- Inference engines
- Opportunistic deduction

V: The Mathematics of Indication, Part 3

- Calculus and time
- Recursion and self-reference
- Autonomous values

Indicational wave-forms
Calculating with autonomous expressions
Imaginary analysis
Paradox resolved
    liar
    Russell
Contradiction maintenance
Fractals

VI: Networks

Distinction networks
Network simplification
Re-entrant forms
Polynomial logic
Wave-trains
Imaginary circuits
Deductive memory

VII: Programming

LISP
Prolog
Losp
Indicative data structures
Context as control
Parallel processing
Connectionistic logic
Optimization
Organic programming
Representing the programmer

VIII: Other Applications

Error models
Systems theory
Number theory
Topology
Machine learning
Philosophy of mathematics
Logic of thought
Focus and meditation
Esoteric schools