The Abstract Method, In General

Gershom Bazerman, S&P/CapitalIQ

"It is natural to attend most to the most set-like aspects of toposes, and to imagine them as derived from set theory, and to do this even without thinking about it. That is how common sense works. Students afflicted with this misunderstanding have trouble escaping the idea that objects are 'really' structured sets and arrows are 'really' structure preserving functions. So they keep looking for the truth 'behind' the category axioms instead of learning to use the axioms. They have trouble learning categorical definitions not because the definitions are too complex but because they believe the axioms must 'really mean' something other than what they say."

–Colin McLarty, "The Uses and Abuses of the History of Topos Theory" (1990)

"Oh. A Monad is a name for IO. Why didn't you tell me?"

"Oh. A Monad is a name for sequencing. Why didn't you tell me?"

"Oh. A Monad is a way to handle errors. Why didn't you tell me?"

–Someone on Reddit, every day, probably as we speak.

Abstraction is Hard To Teach…

…Because we tell people it is unnatural.

The Atomist Fallacy

"To understand a thing is to understand the things it is made out of. To understand those things is to understand the things they are made out of. Once things can no longer be subdivided we understand them fully."

The Atomist Fallacy

The Structuralist Response

- “Objects exist not merely as their constituent elements, but in their arrangement and interaction.”
- There is a ***barrier*** between quantum interactions and classical (Newtonian) mechanics. To move between them is more determined by ***statistics*** than quantum theory directly.
- There is a ***barrier*** between the movement of electrons and assembler. To move between them is determined by the ***structure*** imposed by chip designers.
- There is a ***barrier*** between the objects of a theory, which are just names, and the models of a theory, which are structured by equational laws.
- There is a barrier between assembler and any action executed on a modern computer, not least determined by the operating system itself, which intermediates between userland instructions and raw access to the chip.
- etc.

The (Weak) Structuralist Program

- Whatever the “reality” of mathematical structures, mathematical practice is best conceived of with regards to structures of objects rather than “fundamental” elements.

The (Weak) Structuralist Program

- This is actually not a philosophical claim, but a historical fact with regards to 20th century mathematics.

“[The Axiomatic method] will try, in the demonstrations of a theory, to separate out the principal mainsprings of its arguments; then, taking each of these separately and formulating it in abstract form, it will develop the consequences which follow from it alone. Returning after that to the theory under consideration, it will recombine the component elements, which had previously been separated out, and it will inquire how these different components influence one another. There is indeed nothing new in this classical going to-and-fro between analysis and synthesis; the originality of the method lies entirely in the way in which it is applied.”

–Bourbaki, “The Architecture of Mathematics” (1950)

The Abstract Method in 3 Steps

1. Identify invariant properties over a collection of theories.

2. Systematically ignore the particular properties of objects in each theory. (i.e. ***abstract*** away their particular features, or equally well, quantify over them).

3. Introduce a criterion of identity on the basis of the properties chosen, and thus a new sort of abstract / axiomatically given object, known by the new identity relation, is created.

See: J.P. Marquis, "Mathematical Abstraction, Conceptual Variation and Identity" (2011)

Compare: The "abstract method" refactor

- 1. Take a collection of functions and identify the invariant code between them.
- 2. Systematically "forget" the differences between the functions, leaving only what remains.
- 3. Bind the resultant code to a new name.

A purpose to this

- “Its most striking feature is to effect a considerable economy of thought. The ‘structures’ are tools for the mathematician; as soon as he has recognized among the elements, which he is studying, relations which satisfy the axioms of a known type, he has at his disposal immediately the entire arsenal of general theorems which belong to the structures of that type.” — Bourbaki
- “Each structure carries with it its own language, freighted with special intuitive references derived from the theories from which the axiomatic analysis described above has derived the structure.” — Bourbaki

The Unity of Mathematics

- The relationship of imaginary numbers to the topological structure of the Euclidian plane.
- The duality of combinatorial and categorical accounts of homotopy spaces.
- The many rings: numerical (reals), geometrical (vectors), algebraic (polynomial rings), logical (boolean logic)
- The initially surprising connection between the slope of a line and the area underneath it.
- Toposes as logics, topological spaces, sets, categories, mathematical universes
- Kolmogorov's measure-theoretic account of probability
- Monstrous Moonshine (connects the j-invariant on elliptic curves and the monster group — a finite simple group of order approx. 8 * 10^53).

The Unity of Mathematics (case study)

- Localization or “completion” of an object — iterated complication to produce simpler fields of study.
- Complete N with regards to subtraction, we produce Z (1. Add more points, 2. quotient the duplicates)
- Complete Z with regards to division and we get Q (1. Add more points, 2. quotient the duplicates)
- Complete Q with regards to Cauchy sequences, we get R
- Complete R with regards to roots, and we get C!

The Unity of Mathematics (case study)

- Leinster and Fiore demonstrated that calculations on semirings, such as the semiring of types, make sense if they use complex operations as long as the result may still be written without such operations.
- This should not surprise us, as this is why complex numbers were built to begin with!

The Prism of Mathematics (a poor, suggestive sketch)

- Categorical thinking: Diagram chases, commuting paths, factorization of arrows, presheaf-completions (Yoneda).
- Homotopical thinking: Obstructions, gluings, higher coherences, systems of paths.
- Topological (point set) Thinking: Lattices, semi-decidable (biased) properties
- Harmonic analysis: Rates of growth and oscillation, feedback and cancellation
- Computational thinking: Unfolding processes, termination conditions
- Combinatorial thinking: generating series, counting arguments

Aristotle's program of using philosophy "to lend clarity, directedness, and unity to the investigation and study of particular sciences."

Affirmed in: "Heaviside's 1887 struggle for the proper role of theory in the practice of long-distance telephone-line construction."

– F.W. Lawvere, "Categories of Space and of Quantity" (1992)

Oliver Heaviside

$$\nabla \cdot \mathrm{D} = \rho$$

$$\nabla \cdot \mathrm{B} = 0$$

$$\nabla \times \mathrm{E} = -\frac{\partial \mathrm{B}}{\partial t}$$

$$\nabla \times \mathrm{H} = \frac{\partial \mathrm{D}}{\partial t} + \mathrm{J}$$

- Maxwell’s Equations, but Heaviside’s simplification!

Oliver Heaviside

- Self-Educated (although his uncle was Wheatstone)
- Worked on the Anglo-Danish Telegraph Cable
- Simplified Maxwell's Equations
- Invented the Coaxial Cable and developed Transmission Line Theory
- Developed the Operational Calculus
- Developed the theory of Inductance
- etc.

The use and abuse of abstraction

- Preece (and others) considered induction the enemy — clearly for electrical signals to transmit properly it is necessary to eliminate all magnetic interference that could affect the signal.
- Heaviside argued — adding uniform inductance to a line could reduce distortion and improve the distance of signal transmission.
- People had generalized from submarine cables to all telegraph cables.

The use and abuse of abstraction

- “But all telegraph circuits are not submarine cables, for one thing… The mistake made… was in arguing from the particular to the general. If we wish to be accurate, we must go the other way to work, and branch out from the general to the particular. It is true… that the want of omniscience prevents the literal carrying out of this process; we shall never know the most general theory of anything in Nature; but we may at least take the general theory so far as it is known, and work with that, finding out in special cases whether a more limited theory will not be sufficient, and keeping within bounds accordingly.”
  —Heaviside, “Electromagnetic Induction And Its Propagation” (1887)

“Heaviside formulates what has been my own attitude for the past thirty years: the fact that our knowledge will never of course be complete, and hence no general theory will be final, is no excuse for not using now the most general theory which science can support, and indeed for accuracy we must do so.”

– *F.W. Lawvere, “Categories of Space and of Quantity” (1992)*

Programming as a Mathematical Practice

"To the problem solver, the supreme achievement in mathematics is the solution to a problem that had been given up as hopeless. It matters little that the solution may be clumsy; all that counts is that it should be the first and that the proof be correct. Once the problem solver finds the solution, he will permanently lose interest in it, and will listen to new and simplified proofs with an air of condescension suffused with boredom.

"To the theorizer, the supreme achievement of mathematics is a theory that sheds sudden light on some incomprehensible phenomenon. Success in mathematics does not lie in solving problems but in their trivialization. The moment of glory comes with the discovery of a new theory that does not solve any of the old problems but renders them irrelevant."

—G.C. Rota, *Indiscrete Thoughts*

Programming as a Mathematical Practice

- Mathematicians are like programmers in that they:
- Debug, refactor, accrue and pay down technical debt, take shortcuts, argue endlessly about syntax, and often speak mutually unintelligible languages
- These are social consequences that indicate common elements within the fields themselves

Programming as a Mathematical Practice

- Programmers are like mathematicians in that they:
- Make concerted use of the abstract method.
- Cannot clearly say what programming is or is not.
- Can find virtue in examining the same solution in many ways, some vastly different, some barely perceptibly so.
- Often accidentally rediscover known results.

The Mystery of the Abstract Method

We invent the rules, but we still do not know how the game is played.

The Mystery of the Abstract Method

Programming and Mathematics are both in a sense the art of making things up, consistently. (And making up new meanings of what it is to be consistent).