The Fivefold Way

Category Theory, Physics, Topology, Logic and Computation

David Corfield

Independent researcher
Currently funded by ARIA under the 'Safeguarded AI' program

29 April 2025

Talk structure

- A grand convergence (unification?) is happening between *Category Theory*, *Physics*, *Topology*, *Logic* and *Computation*.
- Its philosophical interest

Grand convergence

Convergence between *Category Theory*, *Physics*, *Topology*, *Logic* and *Computation*.

Grand convergence

Convergence between *Category Theory*, *Physics*, *Topology*, *Logic* and *Computation*.

Some highlights:

- The logic-computation relation from at least the 17th century with Leibniz's *Universal Calculus*, massively extended through the twentieth century.
- Topological semantics for intuitionistic logic, investigated by Tarski and McKinsey from the 1930s.

Grand convergence

Convergence between *Category Theory*, *Physics*, *Topology*, *Logic* and *Computation*.

Some highlights:

- The logic-computation relation from at least the 17th century with Leibniz's *Universal Calculus*, massively extended through the twentieth century.
- Topological semantics for intuitionistic logic, investigated by Tarski and McKinsey from the 1930s.
- Category theory for algebraic topology from 1940s, and algebraic geometry from 1950s. Wiles, Langlands impossible without it.
- Category-theoretic foundations from the 1960s by Lawvere. Topos theory. (Its own form of forcing.) Quantifiers as adjoints.
- Work from the 1970s relating computer science to category theory. Type theory.

Grand convergence

- Physics + topology: from the 1970s, gauge fields = principle bundles

TABLE I. Translation of terminology.

Gauge field terminology	Bundle terminology
gauge (or global gauge)	principal coordinate bundle
gauge type	principal fiber bundle
gauge potential b^k_μ	connection on a principal fiber bundle
S_{ba} (see Sec. V)	transition function
phase factor Φ_{QP}	parallel displacement
field strength $f^k_{\mu\nu}$	curvature
source [a] J^k_μ	?
electromagnetism	connection on a $U_1(1)$ bundle
isotopic spin gauge field	connection on a SU_2 bundle
Dirac's monopole quantization	classification of $U_1(1)$ bundle according to first Chern class
electromagnetism without monopole	connection on a trivial $U_1(1)$ bundle
electromagnetism with monopole	connection on a nontrivial $U_1(1)$ bundle

Tai Tsun Wu, Chen Ning Yang, Concept of nonintegrable phase factors and global formulation of gauge fields, Phys. Rev. D 12 (1975)

Grand convergence

Further highlights:

- Physics and category theory related via TQFT from 1980s, and then later in particular QM and monoidal categories in the work of Bob Coecke and colleagues (Quantum in Pictures, 2023), with connections to quantum computing.
- Monads and comonads from category theory are used in computing to represent effects and contexts from c. 1990, Moggi.

Grand convergence

Further highlights:

- Physics and category theory related via TQFT from 1980s, and then later in particular QM and monoidal categories in the work of Bob Coecke and colleagues (Quantum in Pictures, 2023), with connections to quantum computing.
- Monads and comonads from category theory are used in computing to represent effects and contexts from c. 1990, Moggi.
- Homotopical mathematics: ∞-toposes, later related to physics.
- From 2012, homotopy type theory, closely related to the category-theoretic concept of an ∞-topos, dubbed by Michael Shulman 'The logic of space'.
- Linear homotopy type theory for higher quantum gauge field theory and quantum computing: Sati and Schreiber ('The Quantum Monadology').

The final item sees the interaction of all 5 components of the convergence.

Rosetta Stone, 2009

Baez and Stay call for "a general science of systems and processes", organised by category theory:

Category Theory	Physics	Topology	Logic	Computation
object X	Hilbert space X	manifold X	proposition X	data type X
morphism $f: X \to Y$	operator $f: X \to Y$	cobordism $f: X \to Y$	proof $f: X \to Y$	program $f: X \to Y$
tensor product of objects: $X \otimes Y$	Hilbert space of joint system: $X \otimes Y$	disjoint union of manifolds: $X \otimes Y$	conjunction of propositions: $X \otimes Y$	product of data types: $X \otimes Y$
tensor product of morphisms: $f \otimes g$	parallel processes: $f \otimes g$	disjoint union of cobordisms: $f \otimes g$	proofs carried out in parallel: $f \otimes g$	programs executing in parallel: $f \otimes g$
internal hom: $X \multimap Y$	Hilbert space of 'anti-X and Y': $X^* \otimes Y$	disjoint union of orientation-reversed X and Y: $X^* \otimes Y$	conditional proposition: $X \multimap Y$	function type: $X \multimap Y$

Table 4: The Rosetta Stone (larger version)

John C. Baez, Mike Stay, *Physics, Topology, Logic and Computation: A Rosetta Stone*, 2009.

Harper's Computational Trinitarianism, 2011

> *Computation manifests itself in three forms: proofs of propositions, programs of a type, and mappings between structures. These three aspects give rise to three sects of worship: Logic, which gives primacy to proofs and propositions; Languages, which gives primacy to programs and types; Categories, which gives primacy to mappings and structures.*

Harper's Computational Trinitarianism, 2011

Computation manifests itself in three forms: proofs of propositions, programs of a type, and mappings between structures. These three aspects give rise to three sects of worship: Logic, which gives primacy to proofs and propositions; Languages, which gives primacy to programs and types; Categories, which gives primacy to mappings and structures.

Computational trinitarianism entails that any concept arising in one aspect should have meaning from the perspective of the other two. If you arrive at an insight that has importance for logic, languages, and categories, then you may feel sure that you have elucidated an essential concept of computation—you have made an enduring scientific discovery.

Harper continued

“What I find captivating about computational trinitarianism is that it is *beautiful*! Imagine a world in which logic, programming, and mathematics are unified, in which every proof corresponds to a program, every program to a mapping, every mapping to a proof! Imagine a world in which the *code is the math*, in which there is no separation between the reasoning and the execution, no difference between the language of mathematics and the language of computing.

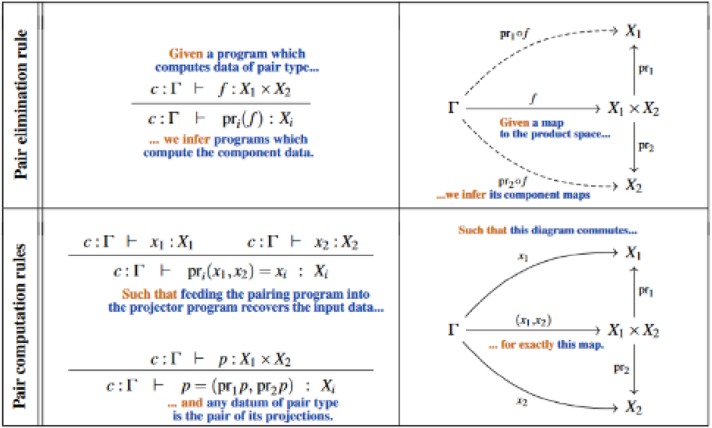
Harper continued

"What I find captivating about computational trinitarianism is that it is *beautiful*! Imagine a world in which logic, programming, and mathematics are unified, in which every proof corresponds to a program, every program to a mapping, every mapping to a proof! Imagine a world in which the *code is the math*, in which there is no separation between the reasoning and the execution, no difference between the language of mathematics and the language of computing. Trinitarianism is the *central organizing principle* of a theory of computation that integrates, unifies, and enriches the language of logic, programming, and mathematics. It provides a *framework for discovery*, as well as analysis, of computational phenomena. An innovation in one aspect must have implications for the other; a good idea is a good idea, in whatever form it may arise. If an idea does not make good sense logically, categorially, and typically..., then it cannot be a manifestation of the divine."

Product

	Programming language (syntax) for **Product types**	**Mathematical denotation** (semantics) of **Product spaces**
Pair type formation rule	Given one data type... ...and another one... $$\frac{\vdash X_1 : \mathrm{Type} \qquad \vdash X_2 : \mathrm{Type}}{\vdash X_1 \times X_2 : \mathrm{Type}}$$ ...we infer their data type of pairs.	Given one space... X_1 ...we infer their product space. $X_1 \times X_2$ X_2 ... and another...
Pair introduction rule	Given a program which computes data of type X_1... ... and a program which computes data of type X_2... $$\frac{c : \Gamma \ \vdash \ x_1 : X_1 \qquad c : \Gamma \ \vdash \ x_2 : X_2}{c : \Gamma \ \vdash \ (x_1, x_2) : X_1 \times X_2}$$ ... we infer a program which computes data of type $X_1 \times X_2$.	Given a map to one space $\Gamma \xrightarrow{x_1} X_1$ $\Gamma \xrightarrow{(x_1,x_2)} X_1 \times X_2$ ...we infer a map to the product space $\Gamma \xrightarrow{x_2} X_2$ and a map to another space

Product



David Jaz Myers, Hisham Sati and Urs Schreiber: Topological Quantum Gates in Homotopy Type Theory

Computational trinitarianism prevails

Logical constructions arise from a 'web of adjunctions' (Lawvere) of elementary operations. (No *tonk*!)

E.g., conjunction is right adjoint to duplication:

$$Hom((C, C), (A, B)) \simeq Hom(C, A \wedge B).$$

This presents the introduction rule, whereas the elimination rule is the counit for the comonad generated by the adjunction.

Computational trinitarianism prevails

Logical constructions arise from a ‘web of adjunctions’ (Lawvere) of elementary operations. (No *tonk*!)

E.g., conjunction is right adjoint to duplication:

$$Hom((C, C), (A, B)) \simeq Hom(C, A \wedge B).$$

This presents the introduction rule, whereas the elimination rule is the counit for the comonad generated by the adjunction.

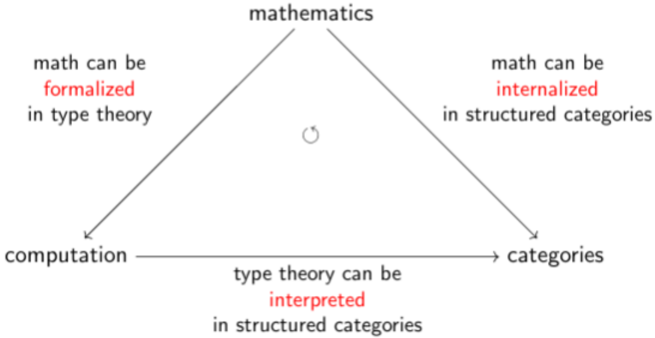
For the left adjoint,

$$Hom((A, B), (C, C)) \simeq Hom(A \vee B, C),$$

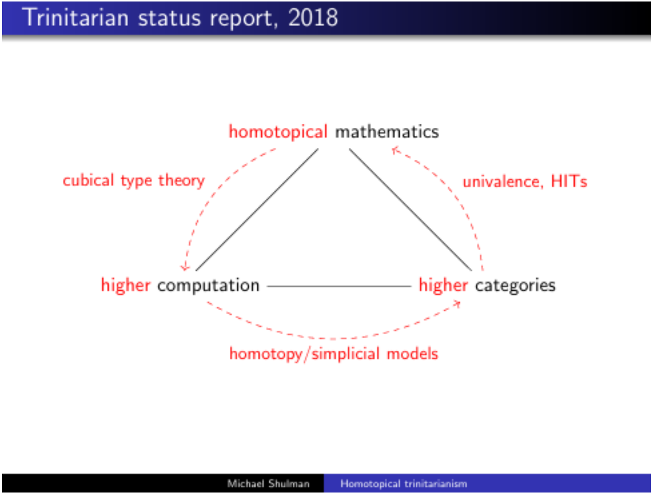
presents the elimination rule, whereas the introduction rule is the unit of the adjunction.
(Polarity of type constructors $\leftrightarrow$ left/rights universal properties.)

Shulman: Homotopical trinitarianism

Homotopical trinitarianism: A perspective on homotopy type theory



Shulman: Homotopical trinitarianism



Homotopical world

- Identity types modelled by path spaces
- Dependent types modelled by spaces fibred over another

Homotopical world

- Identity types modelled by path spaces
- Dependent types modelled by spaces fibred over another
- Path induction allows transfer of structural proofs
- HoTT as internal language of ∞-toposes
- Avoids bad quotienting

Homotopical world

- Identity types modelled by path spaces
- Dependent types modelled by spaces fibred over another
- Path induction allows transfer of structural proofs
- HoTT as internal language of ∞-toposes
- Avoids bad quotienting
- Suited to represent gauge equivalence.
- Cohesive modalities may be added for geometry
- Search is on for a good *directed* HoTT

Sati & Schreiber: Quantum computational trinitarianism

- Higher quantum gauge theory
- Twisted equivariant differential nonabelian generalized cohomology

Sati & Schreiber: Quantum computational trinitarianism

- Higher quantum gauge theory
- Twisted equivariant differential nonabelian generalized cohomology
- Formal language of modal (cohesive) homotopy type theory.
- Takes place in systems of ∞-toposes
- Linear variant for quantum physics
- Relates to *stable* homotopy theory

Sati & Schreiber: Quantum computational trinitarianism

- Higher quantum gauge theory
- Twisted equivariant differential nonabelian generalized cohomology
- Formal language of modal (cohesive) homotopy type theory.
- Takes place in systems of ∞-toposes
- Linear variant for quantum physics
- Relates to *stable* homotopy theory
- Language for topologically-protected hardware verification and software for classically-controlled quantum computing.
- Modalities corresponding to measurement and preparation show that Everett and Copenhagen interpretations are equivalent.

Sati & Schreiber: Quantum computational trinitarianism

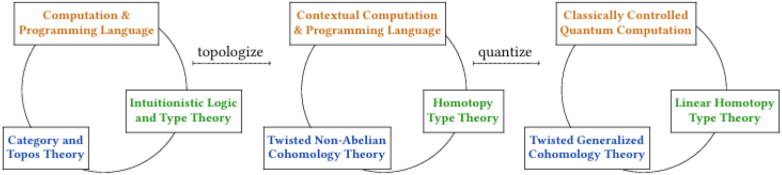
- Higher quantum gauge theory
- Twisted equivariant differential nonabelian generalized cohomology
- Formal language of modal (cohesive) homotopy type theory.
- Takes place in systems of ∞-toposes
- Linear variant for quantum physics
- Relates to *stable* homotopy theory
- Language for topologically-protected hardware verification and software for classically-controlled quantum computing.
- Modalities corresponding to measurement and preparation show that Everett and Copenhagen interpretations are equivalent.

Note: All 5 components of the fivefold way here.

Sati & Schreiber: Quantum computational trinitarianism

The topological quantum trilogy. To appreciate how natural this enhanced programming scheme actually is, notice that we may understand the passage from the left to the right column in the above table as *topologization* followed by *quantization* of the classical *computational trilogy* (**I** below, due to [44, §1] following [42], review in [26, §3]) which puts into mutual relation the theories of
(i) computation, **(ii)** type theory **(iii)** category theory:

- First (in **II**) the *topological computational trilogy* (see [79] with [62, pp. 5-6]) identifies: **(i)** dependent/contextual computation, **(ii)** homotopy type-theory, and **(iii)** homotopy theory;
- and then (in **III**) the *quantum computational trilogy* identifies **(i)** classically-controlled quantum computation, **(ii)** dependent linear homotopy type-theory, and **(iii)** algebraic topology.



I. Classical computational trilogy [44, §1][26, §3]

II. Topological computational trilogy [79][62, pp. 5-6]

III. Quantum computational trilogy [10][70][69, §4.1]

(Topological Quantum Programming in TED-K)

Philosophical interest

Some pointers:

- Revolution: New logic for new mathematics for new physics

Philosophical interest

Some pointers:

- Revolution: New logic for new mathematics for new physics
- Triangulation and ‘realism’

Philosophical interest

Some pointers:

- Revolution: New logic for new mathematics for new physics
- Triangulation and ‘realism’
- Type theory $\leftrightarrow$ phenomenology $+$ ordinary language philosophy

Philosophical interest

Some pointers:

- Revolution: New logic for new mathematics for new physics
- Triangulation and 'realism'
- Type theory $\leftrightarrow$ phenomenology + ordinary language philosophy
- Brandomian expressivism for type theory

Philosophical interest

Some pointers:

- Revolution: New logic for new mathematics for new physics
- Triangulation and 'realism'
- Type theory $\leftrightarrow$ phenomenology $+$ ordinary language philosophy
- Brandomian expressivism for type theory
- HoTT as a structuralist language
- (Since we're in Tübingen) Hegel's Science of Logic, objective and subjective logic

Philosophical interest

Some pointers:

- Revolution: New logic for new mathematics for new physics
- Triangulation and 'realism'
- Type theory $\leftrightarrow$ phenomenology + ordinary language philosophy
- Brandomian expressivism for type theory
- HoTT as a structuralist language
- (Since we're in Tübingen) Hegel's Science of Logic, objective and subjective logic
- Diagrammatics: string diagrams replacing ordinary notation.
- Peirce and category theory
- Other issues in my Modal HoTT book