A Galilean revolution for computing

Unboundedly scalable reliability and efficiency

Conal Elliott

ZuriHac 2023

The Scientific Revolution

- Before the 17th century: conjectures about physical reality.
- Galileo & Newton: theories, experiments, and refutation.

From guessing to knowing

In Galileo's time, professors of philosophy and theology—the subjects were inseparable—produced grand discourses on the nature of reality, the structure of the universe, and the way the world works, all based on sophisticated metaphysical arguments. Meanwhile, Galileo measured how fast balls roll down inclined planes. How mundane! But the learned discourses, while grand, were vague. Galileo's investigations were clear and precise. The old metaphysics never progressed, while Galileo's work bore abundant, and at length spectacular, fruit. Galileo too cared about the big questions, but he realized that getting genuine answers requires patience and humility before the facts.

Frank Wilczek, *The Lightness of Being: Mass, Ether, and the Unification of Forces* (pp. 7-8)

Our computational *crisis* and *opportunity*

- Sequential foundations vs reliability and efficiency. Unsustainable coping strategies:
  - Programming-only languages
  - Testing
  - Model checking, SMT, graph isomorphism
  - Properties
  - Sequential VMs/ISAs & von Neumann bottleneck
- Logic: now formalized and automated

Scalability

- *Efficiency*: correctness (proof).
- *Correctness*: compositionality.
- *Compositionality*: elegance of theorem/specification.
- *Elegance*: simple denotation and homomorphism.

Haskell

- Denotative core: good for correctness, but only “on paper”.
- “Monadic IO” [sic]: sequential and non-denotative: thwarting efficiency and reliability.
- “Functional core, imperative shell”, thwarting scalability.

Beyond Haskell

Haskell types:

- Prevent some bugs.
- Miss the heart of the matter (correctness).

Agda (dependent) types:

- Full specification.
- Simpler and more general.
- Full computation in types/propositions.

What is correct programming like?

- Parsing
- Hardware