Structural Mathematics for Complex Systems

Spencer Breiner

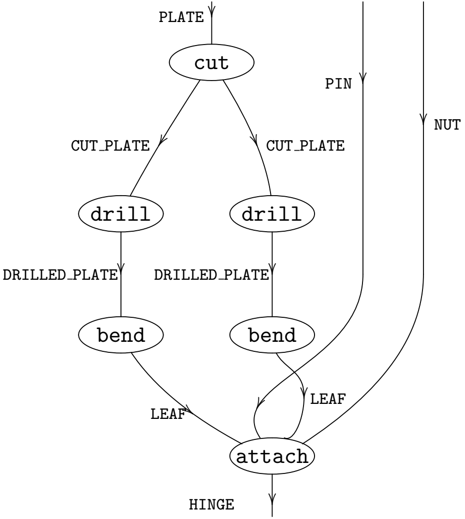
Joint Work with Eswaran Subrahmanian & Al Jones

National Institute of Standards and Technology

September 28, 2015

Outline

1. Complex Systems
2. Structural Mathematics
3. Computational Tools
4. Moving Forward



Why NIST?

Mission: To promote U.S. innovation and industrial competitiveness...

NIST is a branch of the US Department of **Commerce**.

NIST acts as an interface for academia & industry.

Some Interests around NIST:

- Internet of Things
- Cyberphysical Systems
- Systems of Systems
- Global Supply Chain Integration
- Software Security and Specification
- Data Integration
- New Material Design
- Scientific Reproducibility

Some Common Themes

Information Representation

- Rich & varied sensor/actuator data in IoT
- Model-driven design for software

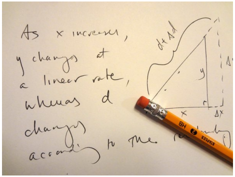
Model Integration

- Dynamic SoS design from off-the-shelf parts
- Data matching & transfer across schemas

Multiple Layers of Structure/Multiple formalisms

- Production line to factory to tech cluster in supply chains
- Micro-, meso- & macro-structure in modern materials

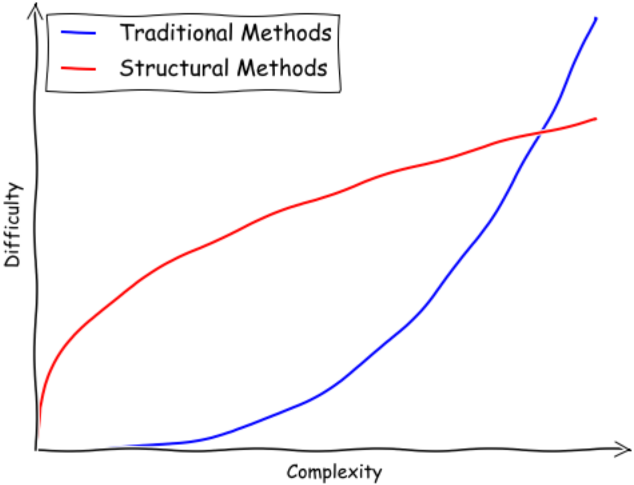
Standards for a New World

1901	1975	2025
 As x increases, y changes at a linear rate, whereas d changes according to the		
Calculus	Logic & Set Theory	Category Theory?

Today, we need a new mathematical foundation for information which:

- Accomodates many formalisms (Matrices, Diff.Eq., Graphs, etc.)
- Scales to address large problems
- Supports evolutionary design & maintenance

Why category theory?



Why category theory?

Information Representation

- Syntactic categories for information modeling
- Presheaves as a context for concrete construction

Model Integration

- Functors for comparing information models
- Colimits for integrating information models
- Sheaves for relating local/global data

Why category theory?

Multiple Layers of Structure

- Methods are composable, with well-defined interactions

$$\texttt{Top} \longrightarrow \varinjlim_i \texttt{Mid}_i$$

Multiple Formalisms

- Developed to bridge gaps in mathematics
- CT is a union of algebraic & geometric methods
- Adjunctions for translating across contexts

Why category theory?

Multiple Layers of Structure

- Methods are composable, with well-defined interactions

$$\texttt{Top} \longrightarrow \varinjlim_i \texttt{Mid}_i$$

$$\texttt{Mid}_i \longrightarrow \varinjlim_j \texttt{Bot}_{ij}$$

Multiple Formalisms

- Developed to bridge gaps in mathematics
- CT is a union of algebraic & geometric methods
- Adjunctions for translating across contexts

Why category theory?

Multiple Layers of Structure

- Methods are composable, with well-defined interactions

$$\begin{tikzcd}
\mathtt{Top} \arrow[r] & \varinjlim_i \mathtt{Mid}_i \arrow[r] & \varinjlim_i \varinjlim_j \mathtt{Bot}_{ij} \\
& \mathtt{Mid}_i \arrow[u, hook] \arrow[r] & \varinjlim_j \mathtt{Bot}_{ij} \arrow[u, hook]
\end{tikzcd}$$

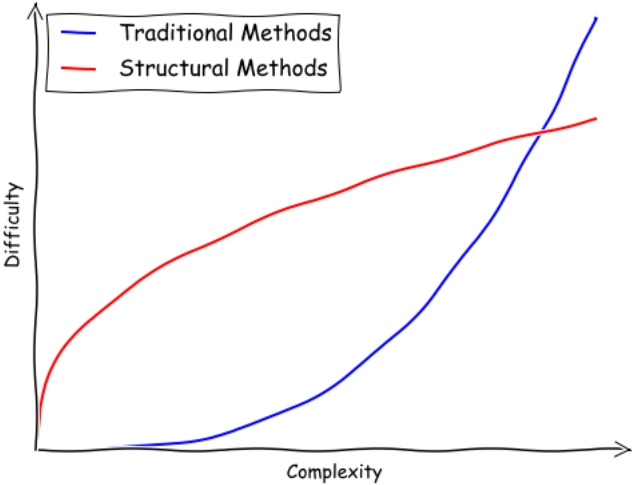
Multiple Formalisms

- Developed to bridge gaps in mathematics
- CT is a union of algebraic & geometric methods
- Adjunctions for translating across contexts

Some other advantages

- People are already using categories without knowing it
  - `List` monad and `map`
  - SQL schemas and database instances
- Graphical formalism
  - UML class diagrams are essentially syntactic categories
  - String diagrams allow easy calculation in SMCs
- Coherent approach to a wide variety of semantic approaches
  - Deterministic, non-deterministic, probabilistic, computational, quantum,...

Why not category theory (yet)?



The Essential Simplicity of Abstract Nonsense

CT is a “big gun” for hard problems.

Broader adoption requires application to *easy* problems.

Strategy: walk before you run

- Posets as categories
- Graphs as functors/presheaves
- Vector space bases as free generation

Hide complexity wherever possible.

The Essential Simplicity of Abstract Nonsense

CT is *not* a "big gun" for hard problems.

Broader adoption requires application to

easy problems.

Strategy: walk before you run

- Posets as categories
- Graphs as functors/presheaves
- Vector space bases as free generation

Hide complexity wherever possible.

The Essential Simplicity of Abstract Nonsense

CT is *not* a “big gun” for hard problems.

Broader adoption requires application to

easy problems.

Most uses of CT involve only 4-5 concepts.

Accessibility is made *much* easier by a new generation of textbooks

Lawvere/Schanuel, Awodey, Spivak,...

Also need (simpler) domain-specific introductions

Computational Tools

Successful CCT tools must leverage this simplicity.

- Can we play with new ideas?

  e.g., “cookbook” examples for user modification

- Can we elide unnecessary/irrelevant details?

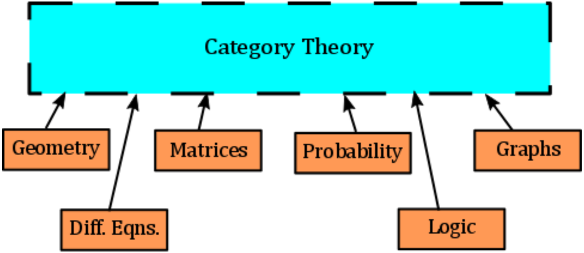
  e.g., type inference or implicit parameters

- Can we use familiar, domain-specific language & notations?

A strong and intuitive graphical user interface is critical!

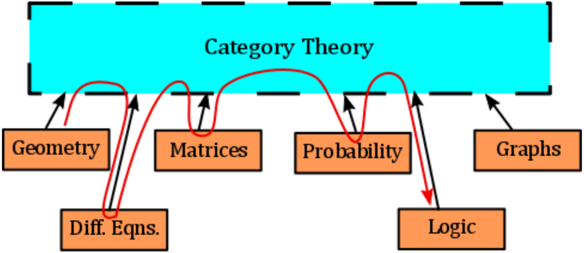
CT as organizational framework

CT can also be the glue binding together other computations.



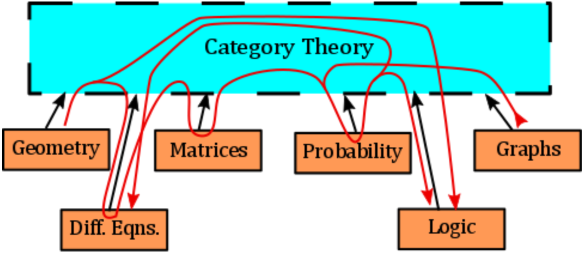
CT as organizational framework

CT can also be the glue binding together other computations.



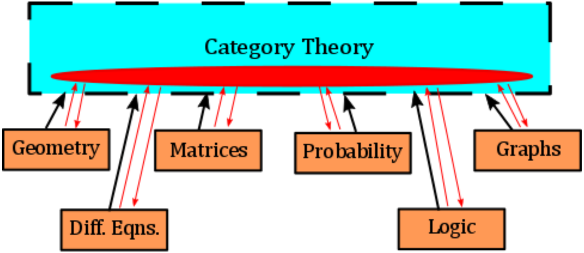
CT as organizational framework

CT can also be the glue binding together other computations.



CT as organizational framework

CT can also be the glue binding together other computations.

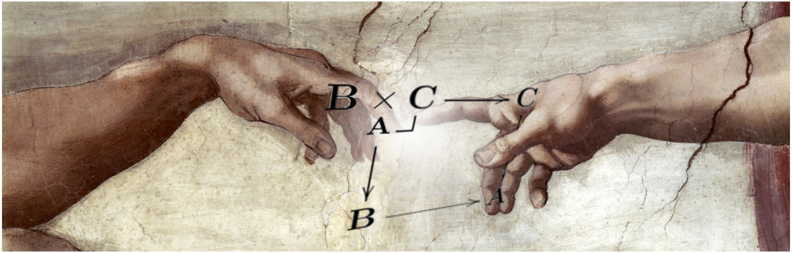


Requires translations to/from existing formats

Leverages existing optimized algorithms

e.g., database join algorithms to compute pullbacks

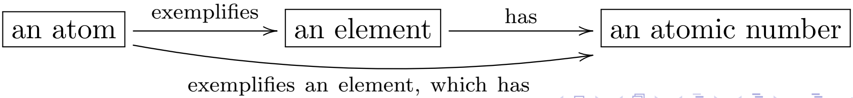
Applications & Outreach



Higher demand will lead to (support for) supply.

Team up with domain experts to map new topics.

Ontology logs (ologs) make categories less scary.


an atom

exemplifies

an element

has

an atomic number

exemplifies an element, which has

Getting there

Providing solid tools will require working together:

- User interface
- Translation to/from existing formats

  SQL, XML, OWL/RDF, Modelica,...
- Categorical algorithms
- Documentation & applications
- Common representation/file format for CT entities

Dividends

For NIST & industry:

- Better formalization of the “soft” sciences
- Easier modularity & integration
- Evolutionary design and maintenance
- More precise graphical language for standards
- Bridge human-readability and computer-readability
- Formal verification & provable guarantees

Dividends

For CT & mathematics at large:

- New problems to study
- Jobs for CT students
- Tools for teaching/learning
- Tools for formal verification
- Unification of pure & applied mathematics