

Computational Category Theory

National Institute of Standards and Technology

Sept. 28 & 29, 2015

Building 101, Lecture Room D

Abstracts

Horn Clause logic of subatomics

Radhakrishnan Balu

Turing-computable Horn clause based logical programming is a very successful technique in expressing classical algorithms. Here, we extend this formalism to the quantum context and discuss algorithms and communication protocols that are commonly found in the literature. The rigorous framework of quantized Horn clauses is established using the machinery of quantum probability. The lattices underlying the quantum logic of Horn clauses characterized and several examples discussed in the proposed language. Finally, open issues in the design of a possible theorem prover to interpret the quantized Horn clauses are discussed.

Structural Mathematics for Complex Systems

Spencer Breiner

Part of NIST's mission is to promote industrial innovation. Looking forward, tomorrow's industry requires a more sophisticated set of tools to address the complexity and dynamism of new mechanisms such as cyber-physical systems and the Internet of Things. Category theory, which has demonstrated its ability to simplify and organize complex problems in mathematics and the sciences, may offer just such a toolbox. A new generation of accessible textbooks makes industrial application of these techniques more realistic, but we still lack the coherent collection of software tools necessary to realize this promise. The

development of this software will benefit not only modern industry, but also category theory itself.

(Module signature/type class/trait/theory presentation) combinators

Jacques Carette

First, from a categorical viewpoint, it is easy to see that module signatures (a la ML), type classes (a la Haskell), traits (a la Scala) and theory presentations (most logics, but also our system MathScheme) all belong to a family of syntactic presentations of certain information. By analysing large code bases, we see a clear need for better mechanisms for building these.

One of the crucial aspect of this information is that in practical applications 1) names matter, but that semantically 2) names don't matter. A straightforward application of categorical ideas deal with #2 easily, but not #1. So, rather than relying on pushouts (as is done traditionally), we instead rely on user-given choices of cartesian morphisms. This gives a much better behaved notion of extension, renaming, "union" and mixin than other approaches.

Using such ideas, couple with extreme modularity [1 concept = 1 module = 1 line of specification] allows us to build a very large set of mathematical theories (1046 at last count) in very few lines (2429 LoC).

Data Landscaping to Support Coordination at Scale

Peter Gates

Pharmaceutical research and development is an endeavor requiring coordination at scale. The high risk together with the inherent scope of product research and development requires that many bets be made and managed across a complex data landscape. The reality of the present day data landscape is more akin to a wild and untamed frontier than to a well-tended garden where information is systematically created, stability and adaptability are in harmony and risks

are understood and managed. With this metaphor in mind we will consider what a data landscape is and how we can go about surveying, standardizing, integrating and evolving that landscape. We coin the phrase data landscaping as a concept that takes an integrated global view of all of the various activities required to provide support for coordination at scale.

Axiomatic Category Theory For Knowledge Representation

Henson Graves

Axiomatic category theory provides a computationally friendly replacement for set theory as a foundation for engineering and science. This claim is based on the use of category theory axiom sets as descriptive models in science and engineering. The language primitives of an engineering modelling language such as SysML can be identified with maps and types (objects). Conversely, the modelling languages can be used to provide a graphical syntax for the topos based languages. Models in these languages can be embedded as axiom sets within constructive topos theory. Constructive axioms can be given that generate a topos with canonical subobjects. The axioms are expressed in a computational logic which is suitable for automated reasoning. This work extends the work of Joachim Lambek on axiomatics for Cartesian closed categories and dogmas. The constructive topos formalism can be used to integrate reasoning with product development in the context of engineering modelling. Automated reasoning from these axiom sets has the potential to solve practical problems. Many problems reduce to whether the theory of the axioms is consistent. Computational toposes can be represented as sections of a sheaf constructed from the Heyting object. The sheaf representation provides a semantics which can be used for simulations of the descriptive models. Meta level characterizations of axiom sets for which all of the valid interpretations are isomorphic are given. These results enable model analysis to be reduced to syntactic properties of axiom sets.

Set-valued functors preserving structure: Presentations, Computations and Implications.

Michael Johnson

Cateno: an approach to computational category theory

Jason Morton

I will discuss the design of an interactive computer algebra system, Cateno, for computational category theory. The emphasis in Cateno is on modeling; it is intended as a "Matlab of category theory" and is targeted at people doing applied work rather than as a proof assistant for category theorists. There is a corresponding design emphasis on performance, accessibility, and working with data. The first models amenable to representation in Cateno are uncertainty models which admit descriptions as monoidal categories. Cateno is implemented in Julia.

Implementing database design categorically

Bob Rosebrugh

Designing models for the syntactic structure of databases is best done using categories. In particular, finite-limit, finite-sum (EA) "sketches" are just right for expressing the both the types and constraints of a database model, especially for the relational paradigm.

The EASIK Java application provides a user-friendly graphical design environment for EA sketches which means that it provides a visualization for such categories. The sketch specifications are XML documents and this is probably EASIK's main interest for Computational CT. The main function of EASIK is exporting a design to a database schema in SQL (the standard relational database language). The application also supports data entry and manipulation. Its most recent version also facilitates definition and export of database views.

FQL: A Functorial Query Language

Ryan Wisnesky

We study the data transformation capabilities associated with schemas that are presented by directed multi-graphs and path equations. Unlike most approaches which treat graph-based schemas as abbreviations for relational schemas, we treat graph-based schemas as categories. A schema \mathbb{S} is a finitely-presented category, and the collection of all \mathbb{S} -instances forms a category, $\mathbb{S}\text{-Inst}$. A functor F between schemas \mathbb{S} and \mathbb{T} , which can be generated from a visual mapping between graphs, induces three adjoint data migration functors,

$$\Sigma_F : \mathbb{S}\text{-Inst} \rightarrow \mathbb{T}\text{-Inst},$$

$$\Pi_F : \mathbb{S}\text{-Inst} \rightarrow \mathbb{T}\text{-Inst},$$

$$\Delta_F : \mathbb{T}\text{-Inst} \rightarrow \mathbb{S}\text{-Inst}.$$

We present an algebraic query language FQL based on these functors, prove that FQL is closed under composition, prove that FQL can be implemented with the select-project-product-union relational algebra (SPCU) extended with a key-generation operation, and prove that SPCU can be implemented with FQL.

A Categorical Approach to Knowledge Management

Ralph Wojtowicz

Categorical logic provides a promising framework for developing knowledge management technologies that complement relational databases, semantic web and logic-based infrastructure. I will discuss joint work with S. Bringsjord and J. Hummel in which we explored applications of sketch theory to knowledge management including representation of context, analogy, inference (via Q-trees of Freyd and Scedrov), and alignment (via the theory of a sketch and Morita equivalence). I will discuss joint work with N. Yanofsky in which we researched quantum algorithms for computing Kan extensions. We found that, quantum computers will not provide a speedup for exact calculation of right Kan ex-

tensions in general or for coproducts. The left Kan extension case depends on whether or not a superior quantum algorithm for computing coequalizers can be found. Finally, I will discuss a project aimed at using category theory to explore transformations between uncertainty models (such as probability and Dempster-Shafer belief states).