Final report for Office of Naval Research Grant N000141310260

David I. Spivak

November 19, 2015

This report will summarize my progress toward the goals of ONR grant N000141310260 ("Categorical Informatics"), which was in effect from 2013/02/01 to 2015/10/31. Throughout this period I have been employed by the Department of Mathematics at the Massachusetts Institute of Technology (MIT). Until March 2013 I was a Postdoctoral Associate, under the guidance of Professor Haynes Miller, and in March 2013 I was promoted to the position of Research Scientist. The Technical Proposal for this grant can be found online at: http://math.mit.edu/~dspivak/informatics/technical_proposal2013.pdf.

Contents

1 Research resulting from this grant

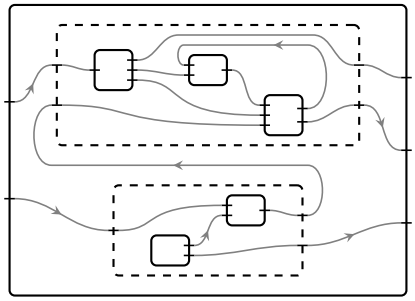
My goals for this grant are discussed in Section I.3 of the Technical Proposal. I will summarize them in a list below, and in the subsequent sections I will discuss the degree to which I was able to accomplish each of them. My stated goals for this grant were to:

- Formalize a notion of hierarchy in information-bearing structures.
- Develop the theory of updates in databases.
- Extend previous work on the use of monads in databases.

I will discuss each of these in turn below, and then I will also discuss some other accomplishments I made during the project period.

1.1 Hierarchy in information-bearing structures

Formalizing the uses of hierarchy, as found in query languages, dynamical systems, and protein materials has constituted a major part of my effort while working on this grant. The categorical tool that turned out to be most relevant was that of operads, which are categories whose morphisms are many-input, one-output. In this context, a morphism is an arrangement for building one thing—a relation, a dynamical system, a protein—as a combination of many smaller things of the same sort.



In the operadic approach to relational algebra, the boxes correspond to table types—i.e., a set of n columns and their types—drawn as a box with n ports, each labeled with its type. A query is a wiring together of such boxes inside of a larger box. Connecting different tables along common columns corresponds to a categorical limit (or JOIN operation), and the outer box represents a final projection (or SELECT clause). The same idea works for dynamical systems (both discrete and continuous). The different semantics correspond to different *algebras* on the same operad.

In the technical proposal for this grant, I also outlined an approach to aggregation using what I called Hierarchical Categories. This approach did not seem to bear fruit. Recently, however, my colleague Ryan Wisnesky came up with nearly the same model independently while trying to formalize the Nested Relational Calculus, an important style of databases used in XML and JSON. It may be that with his help, the hierarchical category idea will play a role in future work.

1.2 Theory of updates

Database updates are commands issues by a user, which take any state and produce a new one. For example, inserting the record (Barack, Obama, 1961) is something that can be done regardless of the current state. Similarly, deleting this record is an update, so it can be done regardless of state; however, note that this update will have no effect if the state does not currently contain this record. Thus database updates

are usually conceived of as functions from the set of possible states to itself. In the category-theoretic setting, one can ask whether updates are in fact *functorial*, and they are. Given a database homomorphism between two states, there is an induced database homomorphisms between their updates. This is true not only for insert and delete updates, but also modify-in-place updates, such as changing Robert to Bob, or increasing everyone's salary by 5%.

Certain updates come with additional structure. For example, if U is an insert update, then for any state S, there is an induced database homomorphism $S \to U(S)$. This is a natural transformation from the identity functor to the update functor. De-duplication updates also come with such a map $S \to D(S)$. On the other hand, if V is a delete update, there is a natural map $V(S) \to S$. The existence of such natural maps has led some categorical database researchers, such as Rosebrugh and Johnson, to consider updates as database homomorphisms, $S \to S'$, but I believe this misses the idea that updates are commands issued independently of state. That said, their work on the view-update problem seems to be workable even in the update-as-functor setting, after just a bit of reworking.

There is a very large category (in a technical sense) of functors C–Set $\to$ C–Set, whereas there is a syntax for, and therefore a countable number of, updates used in practice. Finding a smaller class of updates that is closed under composition is thus an important issue. It appears that the class of polynomial endofunctors $P\colon C\text{–Set} \to C\text{–Set}$ serves in this capacity. Polynomial functors are precisely what we use to capture data migration functors, including ETL processes. It turns out that queries are a particular case (in which we migrate data from our database to a one-table database). We now see updates as another special case, in which we migrate data from our database to itself. This means that queries, updates, and ETL processes all take a common, and composable form.

1.3 Use of monads in databases

Monads can be used to relax the atomicity constraint for relational databases. More precisely, given any monad T on the category of sets, one can consider its Kleisli category Set_T, and define the category of T-states on a database schema C to be the category of functors $C \to \mathsf{Set}_T$. For example, using the powerset monad T, the foreign key columns of a database are filled with sets of IDs from other tables, rather than atomic elements. This is reminiscent of the nested relational calculus, which is based entirely on the powerset monad. As a much more basic instance of this idea, consider the case of *nullable columns*, which are columns in which a value can be NULL. This corresponds to the so-called "maybe" or "option" monad, namely the functor $X \mapsto X + 1$.

In a paper called "Kleisli database instances", I discuss the above idea to use monads in databases. What that paper missed is that different columns should be using different monads. Some columns require atomic data, some allow nullable data, some could allow subsets or probability distributions or general linear combinations of data, etc. It was unclear how to include all this structure. I now understand this much better, as I will discuss below. However, I also realize that by adding so much expressivity, much is lost. For example, instead of having three data migration functors emerge for any functor $F\colon C \to D$ between schemas, there is only one in the above framework. Thus it remains unclear whether this model is viable, especially for information integration tasks.

Let End be the category of finitary endofunctors on Set, a category which includes the finitary monads. We can consider End as a 2-category with one object. There is an 2-embedding $i\colon \mathsf{End} \to \mathsf{Cat}$, sending the unique object to the category Set. Let $\mathcal{S} = \int i$ be the Grothendieck construction applied to this functor. An object in $\mathcal{S}$ is just a set. A morphism $X \to Y$ in $\mathcal{S}$ consists of a finitary endofunctor T together a function $X \to T(Y)$. For example, a function $X \to Y + 1$ is a morphism over the maybe monad. Define a schema in this setting to be a 2-category C over End, so that each morphism is labeled with a monad. An instance on C is then a 2-functor $C \to \mathcal{S}$ over End.

While one data migration functor, pullback along $F\colon C \to D$ does exist, it does not have a left or right adjoint. One way to see this is that the category $\mathcal{S}$ does not have an initial or terminal object. Indeed, if $\emptyset$ is the empty set, we have an isomorphism of categories $\mathrm{Hom}(\emptyset, X) \cong \mathsf{End}$, for any set X.

1.4 Other database research

We have done a good deal of database research that is outside the scope envisioned at the outset of this grant, but which is certainly part of the major thrust of that research agenda. Patrick Schultz and I have discovered a way to much more tightly link a database and an existing programming language. We are currently writing a paper about this, and in the meantime have worked with Ryan Wisnesky to implement the ideas in working code. Wisnesky has a prototype version working, called FQL, in which databases are linked with a Javascript back-end, which can perform arbitrary computations on the data. Queries on such a system are outside the scope of standard SQL and are much more general than database theorists seem to have worked with previously.

We have also used FQL to integrate not only databases but ontologies. Working with researchers from NIST, we went through an example of a supply chain database that was enriched with an OWL ontology, that included a taxonomy of materials. For example, a query for "machines that can drill a 1-inch hole in any kind of steel" should return results for machines that can drill a 1-inch hole in any kind of metal. We were able to capture all this within FQL, and the query results matched NIST's prior published results. This means that instead of stringing together two different tools (database and ontology), FQL could accomplish the same result alone.

Working with some international colleagues, I was able to salvage an old model of databases, called simplicial databases, using a modern logical formulation. The result was a conference paper called Type theoretical databases.

I also formulated SPARQL graph pattern queries, which are often used in with semantic web technologies, using a classical category-theoretic approach called lifting problems. Later, working with Ryan Wisnesky, I realized that graph pattern queries could also be recast as embedded dependencies, which are an important class of constraints used in database research. It turns out that the database technique, called the chase, for repairing non-conforming database states corresponds closely with Daniel Quillen's small object argument. We plan to use this observation to establish more principled variants of the chase procedure, by leveraging Richard Garner's recent work on an algebraic small object argument.

1.5 Other work

In Section 1.1, I mentioned that much of the recent work in my lab has been centered around operads. This has been quite fruitful, leading to a number of papers and several successful grant applications. While I consider only the first of these papers—the one that deals with the relational algebra—to be directly related to the ONR grant, they all have their roots there. This work has also led to a new theorem in rewriting theory, which was discovered jointly with Jason Morton.

I, together with Ravi Jagadeesan, Tristan Giesa, and Markus Buehler, also used operads to produce open-source python software that will construct hierarchical protein materials for use in a molecular dynamics simulator, such as GROMACS or LAMMPS. I have worked a great deal with Buehler's group in the past, resulting in several published articles, but I think this may be our most important contribution to date.

Finally, I worked with Jason Gross and Adam Chlipala to implement a category theory library in Coq, the computer proof assistant. The mathematics is quite layered, and as a result Coq had quite some difficulty handling the complexity. We wrote a paper about how to reduce the burden and make the library more performant.

2 Publications and presentations

Below I will list some publications, presentations, collaborations, and outreach I have been involved with in connection with the ONR grant.

2.1 Book

Spivak, D.I. (2014) *Category theory for scientists*. MIT Press. 486 pages

2.2 Journal articles, refereed conference papers, and technical reports

- Vagner, D.; Spivak, D.I.; Lerman, E. (2015) "Algebras of Open Dynamical Systems on the Operad of Wiring Diagrams". Accepted for publication: *Theory and Application of Categories*. Available online http://arxiv.org/abs/1408.1598.
- Morton, J.; Spivak, D.I. (2015) "A operad-based normal form for morphism expressions in a closed compact category". *Higher-dimensional rewriting and applications*, http://hdra15.gforge.inria.fr.
- Giesa, T.; Jagadeesan, R.; Spivak, D.I.; Buehler, M.J. (2015) "Matriarch: a Python library for materials architecture." *ACS Biomaterials Science & Engineering*, http://pubs.acs.org/doi/full/10.1021/acsbiomaterials.5b00251.
- Spivak, D.I.; Wisnesky, R. (2015) "Relational Foundations for Functorial Data Migration." *Database programming languages*. Available online: http://arxiv.org/abs/1212.5303.
- Brommer D.B.; Giesa T.; Spivak, D.I.; Buehler, M.J. (2015) "Categorical Prototyping: Incorporating Molecular Mechanisms into 3D printing". *Nanotechnology*, article reference: NANO-108127.

- Wisnesky, R.; Spivak, D.I.; Schultz, P.; Subrahmanian, E. (2015) "Functorial data migration: from theory to practice". *NIST Interagency/Internal Report (NISTIR).* Available online: http://arxiv.org/abs/1502.05947.

- Forssell, H.; Gylterud, H.K.; Spivak, D.I. (2016) "Type theoretical databases". *Logical Foundations of Computer Science.* Available online http://arxiv.org/abs/1406.6268

- Spivak, D.I. (2014) "Database queries and constraints via lifting problems." *Mathematical structures in computer science.* Available online: http://arxiv.org/abs/1202.2591.

- Gross, J.; Chlipala, A.; Spivak, D.I. (2014) "Experience Implementing a Performant Category-Theory Library in Coq". *5th conference on interactive theorem proving (ITP'14).* Available online: http://arxiv.org/abs/1401.7694.

- Spivak, D.I.; Wisnesky, R. (2013) "A Functorial Query Language". *Data-Centric Programming workshop (DCP2014).* Available online: http://research.microsoft.com/en-us/events/dcp2014/wisnesky.pdf.

Preprints

- Spivak, D.I.; Schultz, P.; Rupel, D. (2015) "String diagrams for traced and compact categories are oriented 1-cobordisms". *Submitted.* Available online: http://arxiv.org/abs/1508.01069.

- Pérez, M.; Spivak, D.I. (2015) "Toward formalizing ologs: Linguistic structures, instantiations, and mappings". *Submitted.* Available online: http://arxiv.org/abs/1503.08326.

- Spivak, D.I.; Schultz, P.; Wisnesky, R. (2015) "A Purely Equational Formalism for Functorial Data Migration". Available online: http://arxiv.org/abs/1503.03571.

- Spivak, D.I. (2015) "Nesting of dynamic systems and mode-dependent networks". *Submitted.* Available online: http://arxiv.org/abs/1502.07380.

- Spivak, D.I. (2014) "Categories as mathematical models". To appear in *Categories for the Working Philosopher.* Available online http://arxiv.org/abs/1409.6067.

- Rupel, D.; Spivak, D.I. (2013) "The operad of temporal wiring diagrams: formalizing a graphical language for discrete-time processes". *Submitted.* Available online http://arxiv.org/abs/1307.6894.

- Spivak, D.I. (2013) "The operad of wiring diagrams: Formalizing a graphical language for databases, recursion, and plug-and-play circuits." Available online: http://arxiv.org/abs/1305.0297.

2.3 Invited Presentations

U. Mass. Boston (Mathematics colloquium) 2015/10/14;
NIST (Computational category theory workshop) 2015/09/28;
University of Oslo (Department of Informatics) 2015/09/21;
ÉPFL (8 hour mini-course) 2015/09/14 – 2015/09/18;
MIT (LIDS lunch seminar) 2015/06/26;
NIST 2015/06/16;
NIST 2015/06/18;
Foundational Methods in Computer Science 2015/06/06;
Categorical Foundations of Network Theory workshop (ISI Turin) 2015/05/28;
U. Pennsylvania (Complex systems seminar) 2015/04/03;
Pennsylvania State U. (Applied algebra and network theory seminar) 2015/03/18;
MINES ParisTech (International workshop on Design Theory) 2015/01/26;
MIT (Programming languages seminar) 2014/04/15;
IAS (Bar talk) 2014/03/20;
PARC 2014/03/03;
Amgen 2014/03/04;
Oracle 2014/02/28;
UIUC (Topology seminar) 2014/02/25;
Harvard (PL seminar) 2014/02/19;
Carnegie Mellon U. (POP seminar) 2014/01/23;
NIST 2013/06/12;

3 Collaborations, outreach, and transitions

In this section, I will discuss some collaborations that I have taken part in over the grant period (Section 3.1), as well as outreach activities I have performed (Section 3.2). Finally, I will discuss transitions in Section 3.3.

3.1 Collaborations

Below is a list of collaborations, other than those with my postdocs, that have successfully led to papers.

I collaborated with:	**resulting in:**
Members of NIST: Al Jones, Spencer Breiner, and Subrahmanian Eswaran	Two information integration papers, to be published as a result technical report.
Henrik Forssell at U. Oslo	A paper with Forssell's graduate student, which has been submitted to a conference in mathematical logic.
Dylan Rupel at Northeastern	Two papers on wiring diagram operads, both of which have been submitted.
Markus Buehler's group at MIT Civil and Environmental Engineering	Several papers on categorical applications to materials science and engineering.
Adam Chlipala's group at MIT CSAIL	A conference paper on implementing a performant category theory library in Coq, a proof assistant.
Jason Morton at Penn State	A paper presented at a conference on the theory of rewriting systems.
Rajesh Kasturirangan at MIT Laboratory for Information and Decision Systems.	Work in progress on a modular approach to applying the scientific method at scale.
Dmitry Vagner at Duke and Eugene Lerman at UIUC	A paper on composing continuous dynamical systems, accepted for publication in *Theory and Applications of Categories*.

3.2 Outreach

I am interested in disseminating my research more widely to a variety of audiences. To that end, I have participated in the following outreach activities.

I was involved with:	**in which I:**
Hosting a summer graduate student	In the summers of 2014 and 2015, I hosted a graduate student named Dmitry Vagner from Duke University. Our work together resulted in a paper that has been accepted for publication, as mentioned above.
EPFL (École Polytechnique Fédéral Lausanne) in Switzerland	Gave a four-day mini-course in applied category theory to a group of about 20 researchers (professors, postdocs, and graduate students), about half of whom were mathematicians and half were from other sciences.
MIT Undergraduate Research Opportunity Program	I have mentored several undergraduates working to learn and do research in applied category theory.

3.3 Transitions

My former employee, then postdoc, Ryan Wisnesky has started a company, *Categorical Informatics Inc.*, of which I am to be a cofounder. The company will modify the existing

open-source FQL code (which is under a BSD license, with the blessing of MIT), to ready it for commercialization.

My work on operads, wiring diagrams, and modular compositionality led to a need for more research than could be done using ONR funds alone. In December 2013, I was awarded a five-year, $900,000 grant from the Air Force Office of Scientific Research (AFOSR) to study the interaction of agents. In September 2014, I was awarded a three-year $300,000 grant from the National Aeronautics and Space Administration (NASA) to study distributed systems as they relate to the National Air Space.