Categorical informatics:
A functorial approach to data integration

David I. Spivak

November 1, 2015

Part I

Project Summary

Project Summary

Overview: It is proposed that category theory should be further investigated as a potential mathematical foundation for the science of data integration. Category theory is the mathematics of structure, and its focus is on articulating relationships between structures. As such, it is well-equipped to model information frameworks and translations systems, by which data can be migrated and integrated across disparate models. Establishing categorical informatics as a mathematical foundation for information integration has been the subject of intense research at MIT and Harvard over the past five years, and a core framework has been developed. The results have been implemented in working software, which performs many features not usually encountered in database systems, including strong support for data exchange and integration. This proposal seeks to foster the evolution of categorical informatics into a comprehensive platform for high-assurance integration of information.
Keywords: category theory; databases; constraints; view update; aggregation; categorical semantics.

Intellectual Merit: The proposed project is expected to advance knowledge in a transformative way. As an abstract language and reasoning tool, category theory has been revolutionary within mathematics, building bridges between formerly disconnected subfields. It serves as a lingua franca for mathematics, computer science, and informatics, allowing the translation of insights and problem-solving approaches between diverse groups. This bridge-building occurs at multiple scales, from individual teams to large-scale industries. The PI's continuing work with NASA, NIST, Amgen, Janssen, Honeywell, and other governmental and industrial partners speaks to the broad applicability of the categorical informatics paradigm.

The research proposed here will advance knowledge not just within category theory, e.g., through the study of algorithms for computing fibrant replacements in model categories, but more broadly in informatics, where it will provide more predictable information integration techniques, such as chase-like repair procedures with stronger semantics. The success of this research will be measured by the production of software that implements the categorical methods, and that solves problems presented by government and industrial partners.

Broader Impacts: Category theory holds promise to do for data and information what Newton's calculus has done for physics. The ability to integrate information from different frameworks is a major societal issue; for example roughly 40% of enterprise IT budgets—$5B annually—is spent on software of this kind. However, existing tools are invariably ad-hoc, because data integration itself lacks a principled mathematical foundation. Category theory can serve in this capacity, as its purpose is to build bridges between distinct frameworks. Indeed, the PI has written a book, *Category Theory for the Sciences*, which emphasizes bridge-building as it introduces scientists, engineers, and lay people to category theory through applications. Applied category theory is a gateway for learning pure mathematics, providing new access to a sometimes daunting field. Partnering with academics from various disciplines and practitioners from several technology and pharmaceutical companies will create the potential for wide-spread adoption of categorical techniques in a variety of domains.

Contents

Part II

Project Description

1 Introduction

There are at least two somewhat orthogonal aspects to the difficulties encountered in dealing with today's ever increasing volumes of data. One is that, because of the sheer amount being gathered, we need innovative statistical techniques for *summarizing data*—finding meaningful patterns—for classification, determining principle components, et cetera. Another is that data is being produced according to a plethora of different models, and the data can only be exchanged, integrated, and subsequently analyzed when the models are meaningfully coordinated. It is this second problem, that of *coordinating models*, which we propose to address using the mathematics of category theory.

1.1 Background

Data integration is a perennial topic in computer science, being first identified as a problem of significant interest in the 1970s [1], and recently estimated to consume about 40% of enterprise IT budgets, currently around $5 Billion annually [2]. The current mainstream approach to data integration is to view an information-bearing structure, such as a database instance or an ontology, as a model of a theory in some logic, such as first-order logic or description logic. Integration is then formalized as a process by which models of various theories are combined using a variety of constructions (e.g., model merge [3] or the chase [4]). Problems such as data exchange and data migration [5], generic model management [6], schema mapping [7], and extract-transform-load (ETL) [8] can be viewed as special cases of data integration.

Current logic-based approaches to data integration work well in many situations [5], but they have two major limitations. The first is that expressing a domain-specific model (such as an ontology) using formal logic can be cumbersome and subject to numerous errors. This is especially true, for example, when there is a tension between the need to embrace a multitude of valuable, but only partially overlapping perspectives, and the need to form a single, monolithic structure to serve as the domain's entire universe of discourse.

The second limitation involves using logic to facilitate data integration. When different theories are written in different logics, it follows that a logic-based approach may require translations between theories, models, and even logics themselves [6]. Such translations must ensure that the semantics are respected throughout the entire integration process, from actual datasets and documents, through theories and models, all the way to ambient logics. Category theory ameliorates these limitations by providing a single mathematical framework within which theories, models, and logics themselves can be defined, layered, and accordingly related [6].

1.2 Why category theory?

While the name "category theory" evokes notions of classification and categorization, category theory is really the mathematics of modeling itself [9]. The term *model* is a general one—it

applies to all kinds of information-bearing structures—but category theory is equally general: it can be applied throughout science, e.g.,

- in programming language theory, to model functional programs [10];
- in materials science, to model the hierarchy found in protein structures [11];
- in physics, to model the relationship between local and global behavior in various field theories [12];
- in crystallography, to model the symmetry groups of crystal structures [13];
- in mathematics, to model the relationships between geometry and algebra [14].

In fact, category theory has recently been identified by the National Institute of Standards and Technology (NIST) as a potential mathematical foundation for next-generation data integration (e.g., as applied to sensor-actuator data in the Internet of Things, or as applied to supply chains which must account for multiple levels of granularity) [15].

In the category-theoretic approach to modeling, the most important aspect of a model is *how it relates to other models*. In other words, addressing the issue of model coordination—a key problem in the era of big data—is precisely the business of applied category theory. The mathematical terminology for models and the relationships between them is that of *categories* and *functors*. A functor is a mapping between categories,

$$\text{Category}_1 \xrightarrow{\text{Functor}} \text{Category}_2$$

As a high-level connection, a functor *between* two categories is also required to align the multifaceted relationships that exist *within* the categories. In other words, functors are mathematical mappings that respect the integrity of the models involved.

The ability to provide a mathematical basis for articulating the inner structure of models and the relationships between models, as well as the resulting mechanisms for data transformation and integration, has broad applicability to the Information Integration and Informatics program. Applications could include:

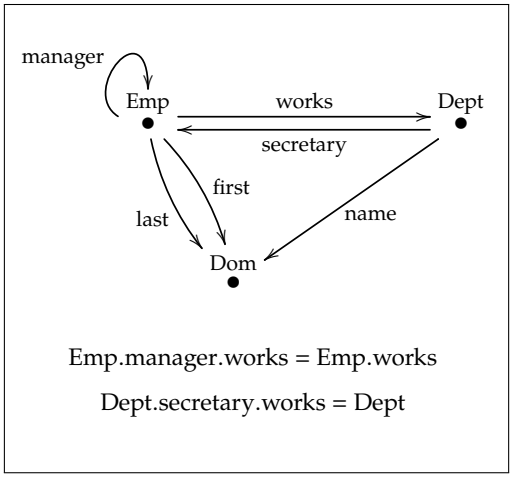
- formalizing the relationships between various data models, such as graph databases, relational databases, ontologies, and knowledge bases;
- giving a mathematical structure and corresponding algorithms for maintaining data provenance, even for data that arrives from various uncoordinated sources; and
- establishing a method by which formalized conceptual models of scientific disciplines can be converted—more or less automatically—into operational tools, e.g., schemas for housing the relevant data.

These applications, all well within the purview of categorical informatics, are not specifically targeted in this proposal. However, it is not unlikely that they will be considered en route while performing the proposed research (see Section 2).

1.3 Previous work

The basic mathematics and computer science founding our categorical approach to data integration have been the subject of intense research at Harvard and MIT over the past five years [16, 17, 18, 19, 20, 21, 22, 23, 24]. In this section, we overview some of the basic technical details of our approach.

A database schema containing primary keys, foreign keys, and path equality constraints is the same as finitely-presented category (Figure 1a), and an instance on that schema corresponds to a set-valued functor (Figure 1b). The collection of all instances on a fixed schema S itself forms a category, denoted S-inst.



(a) A schema is a category S.

Emp				
ID	**mgr**	**works**	**first**	**last**
101	103	q10	Al	Akin
102	102	x02	Bob	Bo
103	103	q10	Carl	Cork

Dept		
ID	**sec**	**name**
q10	102	CS
x02	101	Math

Dom
ID
Al
Akin
Bob
Bo
Carl
Cork
CS
Math

(b) An S-instance is a functor $S \to$ Set.

Figure 1: A database schema and an instance on it

The categorical viewpoint provides a concise formalization of data migration: a mapping between schemas is a functor $F\colon S \to T$, and it induces three data migration operations (indeed, functors), $\Delta_F\colon T\text{-inst} \to S\text{-inst}$, $\Pi_F\colon S\text{-inst} \to T\text{-inst}$, and $\Sigma_F\colon S\text{-inst} \to T\text{-inst}$. Category-theory proves that these migration operations are "the best possible" in a precise sense: they are *adjunctions*. Adjunctions serve as two-way dictionaries between *non-equivalent models* and have many useful formal properties. For example, for all instances I and J we obtain canonical morphisms $J \to \Delta_F(\Sigma_F(J))$ and $\Sigma_F(\Delta_F(I)) \to I$, referred to as the *unit* and the *co-unit* of the adjunctions. These distinguished morphisms relate an instance to its "round-trip" migrations. As another example, adjunctions guarantee useful algebraic equations, such as $\Sigma_F(I \cup J) = \Sigma_F(I) \cup \Sigma_F(J)$. The Δ, Σ, Π data migration operations serve as the primitive operations of the categorical approach to data integration in much the same way that operations such as the chase [4] form a basis for the relational approach to data integration. Sometimes [16], the Δ, Σ, Π data migration operations can be implemented using relational algorithms, and vice versa.

An open-source tool, FQL[1], implements the categorical approach to data exchange in software. The FQL tool provides a graphical viewer for categorical databases and mappings, and allows users to leverage external SQL database engines such as MySQL to perform Δ, Σ, Π data exchange operations efficiently (when possible). The FQL tool has also enabled us to asses the categorical formalism as an approach to real-world data exchange problems. For example, in a joint paper [19] with researchers at NIST, we use the FQL tool to apply the categorical approach and solve a data-integration problem in the additive manufacturing domain.

From the experience gained through previous research and collaboration with a wide variety of scientists and commercial enterprises, we have developed a strong sense of which extensions to the basic theory are needed in order for the categorical informatics framework to provide a truly transformative approach to information integration. It is these extensions that are the topics of this grant proposal.

2 Proposed Work

As described in the previous section (1.3), the basic foundation for a categorical approach to data integration has been established, and it has been implemented in prototype software. However, to be truly practical, research is needed to provide various extensions to the basic theory. The most crucial of these extensions are the following:

1. highly-expressive data integrity constraints,
2. updates and view-updates,
3. nesting and aggregation.

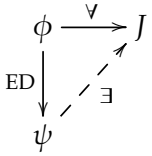
All of these topics have been studied for non-categorical data models (e.g., SQL, XML, RDF, etc.), providing both a wealth of related work from which to draw and a degree of confidence that the extensions identified here are indeed necessary to support the practical deployment of the categorical approach to data integration. We describe these three objectives in Sections 2.1, 2.2, and 2.3 below.

2.1 Proposal topic 1: Constraints

As illustrated in Section 1.3, a database schema containing primary keys, foreign keys, and path equality constraints corresponds to a finitely-presented category S. Expressing additional kinds of data-integrity constraints therefore requires additional structure on S. From the perspective of category theory, the additional structure required is a set of *lifting problems* [20] on the schema. These correspond to what database theorists call *embedded dependencies* [25] (EDs), i.e., formulas of the form $\forall \vec{x}.\phi(\vec{x}) \rightarrow \exists \vec{y}.\psi(\vec{x}, \vec{y})$, where ϕ and ψ are conjunctions of equalities (e.g., $x_1 = x_2$) and memberships (e.g., $R(x_1, x_2)$). An ED is

[1]The FQL (Functorial Query Language) tool is available at `categoricaldata.net/fql.html`.

a constraint on any S-instance J; it says "every case of ϕ found in J can be extended to a compatible case of ψ in J":



One familiar with "Quillen model categories" [26] may think of a set of EDs as a set of generating acyclic cofibrations at which to Bousfield localize the trivial model structure on the co-presheaf category S-inst. A given S-instance J may or may not satisfy (or be *satisfactory* with respect to) this set of EDs. In the language of model categories, J may or may not be fibrant. A set of EDs therefore corresponds to a full subcategory of S-inst, by considering only those satisfactory instances.

EDs are quite general in terms of the kinds of constraints they can encode. From a database point of view, EDs can encode virtually all constraints encountered in practice, such as keys, foreign-keys, join decompositions, and even equalities of conjunctive queries [25]. From a category-theoretic point of view, they can encode the category of models for any limit sketch and more: for example, an ED can enforce that a certain morphism is surjective [20].

When an instance J does not satisfy a set of EDs, one may "repair" J using an algorithm known as the *chase* [4] to obtain a related instance that is satisfactory. The chase repeatedly adds new tuples to J and equates tuples within J until the EDs are satisfied. This algorithm seems to closely correspond to an important procedure in the theory of model categories, called Quillen's *small object argument* [26]. Like the small object argument, the chase lacks any sort of universal property, which makes it suboptimal from the category-theoretic perspective. Recently, Richard Garner [27] produced an algebraic version of Quillen's small object argument which corrects this defect, so it may have concrete applications to database integration problems.

However, the Quillen approach and the Garner approach both have another undesirable property in the database context, which one might call the "duplicated-row" problem. Here is a simplified case: Suppose that S is a schema with a single unary table R. On it, suppose we have a single ED, true $\to \exists x.R(x)$. This ED is satisfied for any instance J having at least one row in R. However, when presented with such an instance *that already satisfies the ED* the small object argument will still repair the instance by adding a new null value to R. This behavior is necessary to ensure functoriality, but it is not desirable from a database perspective because it adds an extraneous null to an already-satisfactory instance.

This presents us with a problem that needs to be solved: how to achieve universality in some form and at the same time remedy the duplicated-row problem. One possible solution might be to somehow "inform" Garner's algorithm that certain lifts already exist for a given instance J. Choosing which row in J is to be designated as the constraint-satisfier is indeed a non-functorial choice; however, once this choice is made, perhaps Garner's algorithm can be run without duplicating these lifts, while still guaranteeing functoriality and universality.

Topic 1 is calling for a universal treatment of the chase, e.g., a variant of Garner's approach, that has the desired database semantics. Besides its utility to the categorical informatics approach to data integration, such a study may also have implications for relational data integration, which relies heavily on the chase.

2.2 Proposal topic 2: View updates

The data in a database is rarely static; it is constantly being updated. Traditionally, database updates are often conceived of as functions, defined according to a schema S, that take old S-instances and produce new S-instances. In our category-theoretic framework, these functions should be replaced by functors: each kind of update—inserting a row, deleting a row, or changing a value—is a functorial mapping

$$\text{Update}\colon S\text{-}\mathsf{inst} \to S\text{-}\mathsf{inst},$$

from the category of S-instances to itself. In fact, an insert is a pointed endofunctor, and a delete is a co-pointed endofunctor. We call these *progressive* and *regressive* updates, respectively.

Formalizing updates as endofunctors appears to be a good approach; however, there are in general a large (uncountable) number of endofunctors $S\text{-}\mathsf{inst} \to S\text{-}\mathsf{inst}$. Since only a countable number of update commands can be expressed in any given language, it is important to limit the class of endofunctors that are to be considered updates. One obvious choice is to use the *polynomial* endofunctors as our limited class, because these are the categorical version of queries [28]. This class leaves out endofunctors like "taking the powerset of a database", which are rarely used as updates in practice. It remains to determine whether all of the updates typically used in practice are indeed polynomial endofunctors.

Once a suitable class of updates is determined, it is crucial to have a good understanding of *view updates*; i.e., how updates interact with queries. For example, a hotel clerk may have a simple view of a large database. Unbeknownst to the clerk, this view is defined by a query, and the clerk interacts with the result, perhaps a single table for reservations at a specific location. When the clerk updates the table, the whole database must be updated accordingly. What makes this a complex issue is that an arbitrary number of integrity constraints may have been imposed on the larger database, and all of them must be maintained. The view-update problem takes many forms, and it has been given many different sorts of solutions [29, 30]. Proposal topic 2 is calling for a formalization of view-updates and relevant theory (e.g., query-rewriting algorithms) to solving the view-update problem with respect to our category-theoretic framework.

Several prior approaches to *relational* view updates use some categorical machinery, and we plan to investigate how this work can be applied in our category-theoretic setting. For example, the approach by Johnson et al. [31] considers view updates as cartesian morphisms with respect to the view functor $S\text{-}\mathsf{inst} \to V\text{-}\mathsf{inst}$. While this notion of update differs from

the standard nomenclature (the updates of [31] are not commands but instead are database homomorphisms [25]), the idea is useful and can be appropriately modified to produce a pointed (resp. copointed) endofunctor on S-inst from one on V-inst, without much difficulty. However, it would be worthwhile to show that this operation preserves the property of being a polynomial. That is, one can ask whether the functor given by lifting a regressive update on V-inst using cartesian morphisms (or a progressive update using op-cartesian morphisms) is a polynomial endofunctor, i.e., an update, on S-inst.

Another possible solution to the view-update problem is to consider complementary views, rather than cartesian morphisms. Here, one cordons off a subset of the database, with which the clerk is to have no contact, guaranteeing that no change will happen there. In this case the view-problem seems to be solvable using adjunctions, because the universality of the solution is restricted to an "active" part of the database. This sort of solution can be greatly generalized using schemas as colimits of other schemas (rather than mere coproducts as suggested in [30]). This adjunction-based approach may also have applications to *mediators* as used in relational data integration [8].

2.3 Proposal topic 3: Nesting and Aggregation

Aggregating data—making sense of it and summarizing it—is one of the most important and common uses of a database. This summarization process is especially important for exchanging information between models, as they may have different levels of granularity.

As the term is typically used, aggregation includes counting the number of employees in a department, or adding up their salaries to find an average. However, aggregation can be considered as a much more general and common phenomenon. Indeed, more complex operations like parameter estimation for curve-fitting is also an aggregation, as is principal component analysis. What all aggregation operations have in common is that data is grouped—gathered into a set, list, or bag—and an operation is performed on the group itself, rather than on individual elements.

Our categorical model, and the software that executes it, already integrates the data store with a real-world programming language, such as Javascript or Python. This programming language can perform parameter estimation as easily as it can perform sums or counts. The necessary, but missing ingredient is to transform flat data into grouped, or nested data, within the categorical model.

Nested data is frequently encountered in practice, especially in data exchange where XML and JSON are de-facto interchange formats; however, the categorical approach to data integration cannot currently represent nested data except by flattening it. Nested data is often formalized using the *nested relational data model* (which was itself inspired by category theory [32]). A schema in the nested relational data model is, for example

```
set (name:str, age:int, Kids:set (name:str, gender:str, Friends: set str))
```

and an instance on that schema is, for example

```
((name:bill, age:30, Kids:{(name:alice, gender:F, Friends: {sue, chuck}),
                           (name:sue, gender:F, Friends: {alice, chuck}),
                           (name:joe, gender:M, Friends: {kim})}),
 (name:bob, age:40, Kids:{(name:chuck, gender:M, Friends: {}),
                           (name:kim, gender:F, Friends: {alice, sue, chuck})}))
```

Schemas in the nested relational model do not contain data-integrity constraints, which must be formalized using nested generalizations of embedded dependencies (EDs) [7]; for example, the constraint that every child has a friend can be formalized as

$$\forall p, k.\ k \in \text{Kids}(p) \to \exists f.\ f \in \text{Friends}(k).$$

Hence, there is interplay between proposal topic 1 (constraints) and this proposal topic (nesting and aggregation), which is calling for

1. a nested analog of finitely-presented categories containing foreign-key constraints,
2. a nested generalization of EDs, e.g., as some sort of lifting problems, as well as theory connecting these extensions to their relational counterparts, and
3. a formalization of aggregation (for example, how to count the number of friends for each child in the above example), appropriate for our category-theoretic setting.

A 2-categorical approach, e.g, using *pro-arrow equipments* [33] may be relevant to this topic. There one could specify certain arrows in a category as functions and others as relations, have access to the fact that every function can be represented as a relation, and also enforce that certain relations were subsets of others. Previous work on using monad algebras [34] for aggregation may also be useful here.

3 Broader Impacts of the Proposed Work

Throughout my career, I have sought to make pure mathematics accessible to a wider audience. I have come to believe the following:

Category theory is the gateway to pure mathematics.

When one knows some basic category theory, one can ask the right questions about sets, groups, spaces, dynamical systems, and chain complexes—not to mention programming languages—and see the connections between them. Category theory serves as a foundation not just for the mathematics itself, but for the way we actually think and speak most clearly about it.

If this vision of category theory as a gateway is correct, then an attempt to disseminate pure mathematics should begin with the broad dissemination of category theory. To that end, I wrote a book called *Category Theory for the Sciences*, published in 2014 by MIT Press [35].

I have also given colloquium talks on the applications of category theory at the University of Oregon and the University of Massachusetts, Boston, and a four day (two hours per day) mini-course at the École Polytechnique Fédéral de Lausanne in Switzerland.

The research I propose here will continue to facilitate the dissemination of category theory, because it focuses on a concrete problem, namely that of coordinating models and translating information between them. Bringing the math down to earth with FQL, our open-source data integration tool, serves an additional pedagogical purpose. That is, it enables users to play with the abstract mathematics: one can input finitely presented categories and set-valued functors, and see them simply displayed as databases. One can input functors, see them displayed as maps between schemas, and then watch as the data is migrated from one database to the other.

Focusing on concrete problems, both in talks and in research, attracts the attention of a broad audience. It has generated informal and formal collaborations with academics in robotics, neuroscience, computer science [16, 37], manufacturing [19], and materials engineering [11, 40]. I also collaborate informally and formally with several international colleagues, e.g., with Stephen Molloy at the European Spallation Source and with Henrik Forssell at the University of Oslo [41].

I also often work with people outside of academia, e.g., from pharmaceutical and biotech companies (Amgen, Janssen, Foundation Medicine), from technology companies (Honeywell), from software companies (Microsoft and others), from government organizations (NIST). In these interactions, both partners typically learn a great deal. For example, in my work with Honeywell on a NASA grant, I have become much more acquainted with the assets and procedures that assure safe separation of airplanes in the National Air Space, and similarly people from Honeywell and NASA have become much more acquainted with the language and techniques of category theory.

One of my postdocs has now left academia to start a category-theory based information integration company, called *Categorical Informatics Inc*. The company is currently enrolled in an upcoming cohort of the NSF Innovation Corps Team program, where it will look for the best fit between products it could make and markets that may have interest. Any work we do for the IIS grant will be published, and any code will be made open-source (using a BSD license), at which point the company can convert it into a salable product that will have directly impacts on society.

The work required for this grant lends itself to a vertical integration of researchers from undergraduate and graduate students to postdocs to professors. For example, since arriving at MIT in 2010, I have worked independently with fourteen (14) undergraduate students for a total of nineteen (19) semesters—I worked with several students more than once—as their mentor in the Undergraduate Research Opportunity Program (UROP). I had these students work on small models of big problems, and more than once their work has influenced or directly led to publications ([35, 37, 38, 39]). I have also worked with a graduate student from Duke for two summers, resulting in an accepted paper [36]. I have hired postdocs from

around the world (including a Greek woman and a Venezuelan man), who were trained in pure category theory or homological algebra, and who have subsequently learned to apply their knowledge to real-world problems.

I will certainly continue this kind of community building and dissemination of pure mathematics through the gateway of category theory, and being awarded this grant would no doubt aid me in that effort. Moreover, the abstract mathematics we produce will have concrete applications to information exchange problems. It will thus be directly useful to scientists and engineers, as it will improve their ability to coordinate models and translate knowledge between disparate domains. The products of our research will similarly be useful in society at large, because in order to tackle large-scale problems such as climate change, space exploration, etc., an emphasis on coordinating scientific models and translating results is strongly needed. Applied category theory, and the proposed categorical informatics project in particular, will continue to help lay the foundation for a truly interdisciplinary and cooperative society.

4 Results from prior NSF support

The only prior NSF support I have had was through the I-Corps Site program, a $1,500 micro-grant distributed by MIT's Venture Mentoring Service. Through this grant, a new company called *Categorical Informatics Inc*, recently spun out of MIT, was introduced to the customer-discovery process. This provided a bit of the background needed for the company to successfully commercialize the technological results of prior categorical informatics research at MIT.

The company has recently been selected to participate again at the next level, in the I-Corps Team program in New York City. There, the team's entrepreneurial lead will be coached through a process of interviewing over 100 contacts in a variety of companies, with whom he will discuss the potential market for categorical approaches to data integration. The impacts on society have yet to be seen, but certainly the process will lead to an exposure of more companies to the idea that category theory can serve as a new mathematical approach to information integration.

Intellectual Merit: Information integration is a perennial topic in computer science, being first identified as a problem of significant interest in the 1970s [1]. There are tools, such as extract-transform-load (ETL) tools, that perform certain information integration tasks, but these tools are invariably ad-hoc [1]. As a natural mathematical framework for describing models and the relationships between them, category theory provides *principled* techniques for solving the information integration problems that plague scientists, engineers, and commercial enterprises. Although category theory has revolutionized several areas of computer science—for example, functional programming [10]—category theory is only beginning to

be applied to problems in information integration. It is this application that we addressed using the I-Corps Site grant.

Broader Impacts: According to the market intelligence firm IDC, 40% of all enterprise IT budgets are dedicated to solving information-integration problems, and total annual spending of information-integration software alone is over $5 billion [2]. Information-integration problems are currently solved using a combination of manual data manipulation and custom software, all based on the relational model of data. The categorical model we have developed has been formally proven to provide more accurate solutions to certain information-integration problems than the relational model and to solve certain information-integration problems that the relational model cannot [16]. And because relational databases are a special case of categorical databases, our approach can easily interoperate with existing technologies. Hence, practical software based on our categorical model has the potential to significantly ease the burden of challenging real-world IT problems. In addition, the success of an information-integration tool based on category theory may inspire other efforts to leverage category theory for data management tasks. The specific results of our work on the I-Corps Site program was that we began our exploration of the market need for our information integration tool, to be continued in the I-Corps Team program early next year.

Part III

References Cited

References cited

[1] Fagin, R.; Kolaitis, P.; Popa, L. (2003) *Data Exchange: getting to the core*. Principles of Database Systems (PODS), ACM.

[2] Bernstein, P.A.; Hass, L.M. (2008) "Information integration in the enterprise". *Communications of the ACM* Vol 51, No. 9.

[3] Melnik, S. (2004) *Generic Model Management: Concepts and Algorithms*. Lecture Notes in Computer Science, Springer.

[4] Deutsch, A.; Nash, A.; Remmel, J. (2008) *The chase revisited*. Principles of Database Systems (PODS), ACM.

[5] Fagin, R.; Colitis, P.; Miller, R.; Popa, L. (2003) *Data Exchange: semantics and query answering*. Proceedings of the International Conference on Database Theory (ICDT), Spinger Publishing.

[6] S. Alagic, P. Bernstein. (2001) *A Model Theory for Generic Schema Management*. Proceedings of the International Symposium on Database Programming Languages (DBPL), ACM.

[7] A. Fuxman et al. (2006) *Nested Mappings: Schema Mapping Reloaded*. Very Large Databases (VLDB), ACM.

[8] Doan, A.; Halevy, A.; Ives, Z. (2012) *Principles of Data Integration*. Morgan Kaufmann Publishing.

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

[10] Moggi, E. (1991) *Notions of computation and monads*. Information and Computation Journal Volume 93 Issue 1.

[11] 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.

[12] Sati, H.; Schreiber, U. (2011) *Mathematical Foundations of Quantum Field Theory and Perturbative String Theory*. American Mathematical Society.

[13] Fichtner, K. (1980) "On groupoids in crystallography". *Match 9*, pp. 21 – 40.

[14] Eilenberg, S.; MacLane, S. (1945) "General theory of natural equivalences." *Trans. Amer. Math. Soc.* 58, pp. 231 – 294.

[15] Breiner, S. (2015) "Structural Mathematics for Complex Systems" Available online: http://www.appliedcategorytheory.org/wp-content/uploads/2015/10/Breiner-Structural-Mathematics-for-Complex-Systems.pdf

[16] Spivak, D.I.; Wisnesky, R. (2015) *Relational Foundations for Functorial Data Migration*. Proceedings of the International Symposium on Database Programming Languages (DBPL), ACM.

[17] Wisnesky, R. (2013) "Functional query languages with categorical types". PhD Thesis, Harvard.

[18] Spivak, D.I. (2012) "Kleisli database instances". Available online: http://arxiv.org/abs/1209.1011.

[19] 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.

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

[21] Spivak, D.I.; Kent, R.E. (2012) "Ologs: a categorical framework for knowledge representation". *PLoS ONE* 7(1): e24274. doi:10.1371/journal.pone.0024274.

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

[23] 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

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

[25] Abitebould, S.; Hull, R.; Vianu, V.. (1995) *Foundations of Databases*. Addison-Wesley Longman Publishing.

[26] Quillen, D.G. (1967) *Homotopical Algebra*. Lecture notes in mathematics, No. 43. Springer-Verlag.

[27] Garner, R. (2009) "Understanding the small object argument", *Appl. Categ. Structures* 17 (3), pp. 247–285.

[28] Spivak, D.I. (2012) "Functorial Data Migration". *Information and Communication*. Vol 217, pp. 31 – 51. Available online: http://arxiv.org/abs/1009.1166

[29] Hofmann, M.; Pierce, B. C.;Wagner, D. (2011) "Symmetric Lenses." In *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*.

[30] Bancilhon, F.; Spyratos, N. (1981) "Update semantics of relational views." *ACM Transactions on Database Systems (TODS)* 6.4: 557-575.

[31] Johnson, M.; Rosebrugh, R.; Wood, R. J. (2012). Lenses, fibrations and universal translations. Mathematical Structures in Computer Science, 22(01), 25-42.

[32] Wong, L. (1994) *Querying Nested Collections*. Ph.D. thesis, University of Pennsylvania.

[33] Shulman, M. (2008) "Framed bicategories and monoidal fibrations." *Theory and Applications of Categories* 20, No. 18, 650–738.

[34] Lellahi, S.; Tannen, V. (1997) *A Calculus for Collections and Aggregates*. Proceedings of the International Conference on Category Theory and Computer Science (CTCS). Springer-Verlag.

[35] Spivak, D.I. (2014) *Category Theory for the Sciences*. Cambridge: MIT Press. 486 pages.

[36] 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

[37] 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

[38] 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

[39] 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

[40] Giesa, T.; Spivak, D.I.; Buehler, M.J. (2012) "Category theory based solution for the building block replacement problem in materials design". *Advanced Engineering Materials*. DOI: 10.1002/adem.201200109

[41] Forssell, H.; Gylterud, H.K.; Spivak, D.I. (2015) "Type theoretical databases". To appear in: *Logical Foundations of Computer Science*. Available online http://arxiv.org/abs/1406.6268

Part IV

Biographical Sketch

David I. Spivak
(Principal Investigator)

Department of Mathematics
Massachusetts Institute of Technology
77 Massachusetts Avenue
Building E17 Room 417
Cambridge, MA 02143
dspivak@mit.edu

Professional Preparation

University of Maryland, College Park, MD: BS 1996
University of California, Berkeley, CA: PhD 2007

Appointments

Since 2013: Research Scientist, Massachusetts Institute of Technology
2010 – 2013: Postdoctoral Associate, Massachusetts Institute of Technology
2007 – 2010: Visiting Assistant Professor, University of Oregon

Five Related Products

- Spivak, D.I. (2012) "Functorial Data Migration". *Information and Communication*. Vol 217, pp. 31 – 51.
- Spivak, D.I. Wisnesky, R. (2015) "Relational Foundations for Functorial Data Migration". *Proceedings of the International Symposium on Database Programming Languages (DBPL)*, ACM.
- Spivak, D.I. (2014) *Category Theory for the Sciences*. Cambridge: MIT Press. 486 pages.
- Spivak, D.I. (2014) "Database queries and constraints via lifting problems." *Mathematical structures in computer science*.
- Wisnesky, R.; Spivak, D.I.; Schultz, P.; Subrahmanian, E. (2015) "Functorial data migration: from theory to practice". *NIST Interagency/Internal Report (NISTIR)*.

Five Other Significant Products

- Giesa, T.; Spivak, D.I.; Buehler, M.J. (2012) "Category theory based solution for the building block replacement problem in materials design". *Advanced Engineering Materials*. DOI: 10.1002/adem.201200109
- Spivak, D.I. (2014) "Categories as mathematical models". To appear in *Categories for the Working Philosopher*.
- Spivak, D.I.; Kent, R.E. (2012) "Ologs: a categorical framework for knowledge representation". *PLoS ONE* 7(1): e24274. doi:10.1371/journal.pone.0024274.

- Forssell, H.; Gylterud, H.K.; Spivak, D.I. (2015) "Type theoretical databases". To appear in: *Logical Foundations of Computer Science*.
- 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)*.

Five Synergistic Activities

- I taught "Category theory for scientists" in Spring 2013 at MIT, a first-of-its-kind course on applied category theory. The 18 students were from math, materials science, computer science, neuroscience, and other fields. The course textbook *Category Theory for the Sciences* has led to interest from people of a wide variety of backgrounds, both geographically and in terms of mathematical sophistication.
- I hired and worked with Ryan Wisnesky to create a software tool, FQL, which can be used to teach category theory, including left and right Kan extensions as "data migration functors". A company, *Categorical Informatics Inc.* is now being spun out of MIT, to commercialize this product.
- I have hired researchers from a variety of background as postdocs. Out of four postdocs hired to date, one is a woman, one is latino, and another is half latino.
- Since 2010, I have mentored fourteen undergraduate students of all backgrounds on nineteen research projects in applied category theory. Several of these projects resulted in published papers. I have also had a visiting graduate student for the past two summers, resulting in a published paper.
- I co-organized the first Computational Category Theory conference at NIST in 2015 and the "Workshop on topology and abstract algebra for biomedicine" at the Pacific Symposium on Biocomputing (PSB) 2016.

Collaborations and other affiliations

Collaborators and Co-Editors (15)

Spencer Breiner (NIST), Markus Buehler (MIT), Adam Chlipala (MIT), Subrahmanian Eswaran (CMU), Henrik Forssell (University of Oslo), Tristan Giesa (MIT), Jason Gross (MIT), Hakon R. Gylterud (Stockholm University), Al Jones (NIST), Eugene Lerman (UIUC), Marco Pérez (University of Mexico), Dylan Rupel (Notre Dame), Patrick Schultz (MIT), Dmitry Vagner (Duke), Ryan Wisnesky (Categorical Informatics Inc).

Graduate Advisors and Postdoctoral Sponsors (4)

Tom Graber, Jacob Lurie, Daniel Dugger, Haynes Miller.

Thesis Advisor (1)

Peter Teichner (University of California, Berkeley).

Part V

Postdoctoral Mentoring Plan

Postdoctoral Mentoring Plan

I will mentor each postdoc I hire for this grant as follows.

I plan to meet with the postdoc at least once a week, or more. We may discuss the project as well as other relevant mathematical questions and ideas. We will also discuss questions that the postdoc has about their career, work-life balance, and so on. We will also periodically discuss their progress and any outstanding issues. In the background, I will promote responsible professional practices at all times.

In accordance with the topics laid out in the project description, there are several concrete problems that the postdoc can work on. I will be available to discuss possible solutions, to find new avenues when there are roadblocks, to evaluate results, etc. In short, we will collaborate on the proposed research problems. I will also encourage the postdoc to collaborate with other members of my group, as well as with other researchers at MIT, both inside and outside of the math department. I am happy to fund travel for the postdoc to collaborate with researchers at other universities as well, or to bring in visitors.

The postdoc may also work independently on some problems. I will help them decide if their research is publishable, and when it is I will review and suggest improvements to draft articles. I will also suggest suitable venues for publication and assist in interpreting referee's comments. Similarly, I will have the postdoc help me in writing grant proposals, or I will help them write their own.

The postdoc may supervise undergraduate researchers or be on part of a supervisory team for visiting graduate students. This will give the postdoc practice in discussing the proposed research with people from a diverse background. This will also be the case when we invite industrial partners, such as data quality personnel at pharmaceutical companies, to explain real-life database integration problems: the postdoc will learn to communicate to a broad range of interested parties.