Minimal Schemas for a Category

(extended abstract)

Liang Ze Wong

July 15, 2019

1 Introduction

This paper is motivated by the question: *When is a category C the category of elements $\int F$ of a functor $F\colon S \to \mathbf{Set}$?* Via the correspondence between functors $F\colon S \to \mathbf{Set}$ and discrete opfibrations $P\colon C \to S$, this is equivalent to the question: *When do we have a discrete opfibration $P\colon C \to S$?* In either case, we call S a **schema** for C.

We always have a trivial schema for C, given by $S = C$ (and $P = 1_C$). In other words, every C is a schema for itself. But we would like to know if there are non-trivial solutions to this problem that are smaller than C in a suitable sense. Even better, we would like to find the smallest such schema, or the **minimal** schema, for C.

To motivate this question, recall from [Spi12] that a functor $F\colon S \to \mathbf{Set}$ is precisely a database with schema S. For $f\colon s \to t$ in S, we write F_s, F_t for the sets $F(s)$ and $F(t)$, and F_f for the function $F(f)\colon F(s) \to F(t)$. Given any $F\colon S \to \mathbf{Set}$, we may form its **category of elements** $\int F$ whose objects are all pairs (s, x) where $s \in S$ and $x \in F_s$, and whose morphisms $(s, x) \to (t, y)$ are all arrows $f\colon s \to t$ in S such that $F_f(x) = y$.

The category of elements allows us to encode a database $F\colon S \to \mathbf{Set}$ as a category. Our question thus asks when a given category C encodes the structure and information of a database $F\colon S \to \mathbf{Set}$.

Acknowledgements

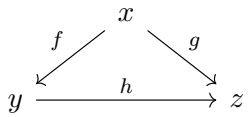
An earlier version of this abstract was submitted to Applied Category Theory 2019, and presented as a poster. We would like to thank the anonymous referees for their comments, which have helped streamline this paper.

We note also that the question, "Does there exist a terminal surjective discrete (op)fibration out of C?" has already been asked by David Spivak [Spi14]. This paper provides a partial answer to that question.

2 Constructing a schema for C

We first give a quick overview of coslice categories, which feature prominently in the main result. For a given object x in a category C, the *coslice category* x/C is the category whose

objects are morphisms in C with domain x, and whose morphisms are commuting triangles of the form:



Any $f\colon x \to y$ in C induces a functor $- \circ f\colon y/C \to x/C$. Any functor $F\colon C \to D$ also induces a functor $x/C \to Fx/D$. We note some results concerning these functors and isomorphisms of coslice categories:

Lemma 2.1. *Let $\varphi\colon x/C \cong y/C$ be an isomorphism of coslice categories. For $f\colon x \to u$, let v be the codomain of $\varphi(f)$ (so we have $\varphi(f)\colon y \to v$). Then φ induces an isomorphism $u/C \cong v/C$ making the following diagram commute:*

$$\begin{array}{ccc} u/C & \xrightarrow{\cong} & v/C \\ {\scriptstyle -\circ f}\downarrow & & \downarrow{\scriptstyle -\circ\varphi(f)} \\ x/C & \xrightarrow[\cong]{\varphi} & y/C \end{array}$$

Lemma 2.2. *Let $P\colon C \to S$ be a functor. Then P is a discrete opfibration if and only if the induced functor $x/C \to Px/S$ is an isomorphism of categories for every $c \in C$.*

As an immediate consequence, we have a necessary condition for two objects of C to be identified by a discrete opfibration:

Corollary 2.3. *Let $P\colon C \to S$ be a discrete opfibration, and suppose we have $x, y \in C$ such that $Px = Py$. Then x/C and y/C are isomorphic.*

The main theoretical result of this paper is that the converse holds under certain conditions on C:

Theorem 2.4. *Let C be such that each x/C has no non-identity automorphisms. Then there exists a discrete opfibration $P\colon C \to S$ such that $Px = Py$ if and only if $x/C \cong y/C$.*

The proof of requires the following lemma:

Lemma 2.5. *Let C be such that each x/C has no non-identity automorphisms. Then:*

1. *Any isomorphism $x/C \cong y/C$ sends 1_x to 1_y;*
2. *For x, y such that $x/C \cong y/C$, there is a* unique *isomorphism $\varphi\colon x/C \to y/C$.*

Proof. For 1, we first observe that the hypothesis on C implies that C is skeletal. Thus each x/C has a unique initial object 1_x, and initial objects are preserved by isomorphisms.

For 2, if $\varphi, \psi\colon x/C \to y/C$ were two distinct isomorphisms, then $\psi^{-1}\varphi$ would be a non-identity automorphism of x/C. □

Proof of Theorem 2.4. We will define a category S and a functor $P\colon C \to S$ with the desired properties. The objects of S are isomorphism classes of coslice categories, denoted $[x/C]$ for $x \in C$. For morphisms, let

$$S([x/C],[y/C]) := \coprod_{\substack{y' \in C \text{ s.t.} \\ y'/C \cong y/C}} C(c, y').$$

The identity on $[x/C]$ is $1_x \in C(x,x) \subset S([x/C],[x/C])$. To compose $f \in S([x/C],[y/C])$ with $g \in S([y/C],[z/C])$, suppose $f \in C(x,y')$ and $g \in C(y,z')$, where $y'/C \cong y/C$ and $z'/C \cong z/C$. Let $\varphi\colon y/C \to y'/C$ be the unique isomorphism between coslices, and let u be the codomain of $\varphi(g)$ (so we have $\varphi(g)\colon y' \to u$). Then $u/C \cong z'/C \cong z/C$ by Lemma 2.1, so we may define the composite to be

$$g \circ f := \varphi(g) \circ f \in C(x,u) \subset S([x/C],[z/C]).$$

Item 1 of Lemma 2.5 ensures that composition is unital, while Item 2 ensures that it is associative. We thus have a category S.

It is easy to see that there is a functor $P\colon C \to S$ sending x to $[x/C]$ and $f\colon x \to y$ to itself, treated as an element of $S([x/C],[y/C])$. By construction, $Px = Py$ precisely when $x/C \cong y/C$.

Given $[x/C] \in S$ and x' such that $Px' = [x/C]$ (so $x'/C \cong x/C$), and $f \in C(x,y') \subset S([x/C],[y/C])$, we have a unique map $\varphi(f)\colon x' \to u$ lifting f, where $\varphi\colon x/C \cong x'/C$. Thus $P\colon C \to S$ is a discrete opfibration. □

Remark 2.6. *The hypotheses on C are only used to show that composition is unital and associative. In the absence of these hypotheses, one would need to coherently choose isomorphisms between coslice categories in order to have unital and associative composition. It is unclear if this can always be done, although the author believes so. After all, not all categories of elements satisfy the hypotheses of the theorem.*

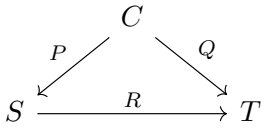
3 Minimality of the schema

We now prove that the schema S constructed in the previous section is minimal in a precise sense. We begin by defining an appropriate category in which $P\colon C \to S$ is minimal.

Definition 3.1. Let C be a category. The **category of schemas** $C_{\mathbb{S}}$ under C is the full subcategory of $C/\mathbf{Cat}$ whose objects are surjective discrete opfibrations $P\colon C \to S$.

Note that we are only interested in discrete opfibrations that are surjective (i.e. surjective on objects). Since we are interested in schemas for C, we lose nothing by throwing away objects of S whose pre-images in C are empty. Also, surjectivity allows us to prove:

Lemma 3.2. *Suppose we have a commuting triangle of functors*



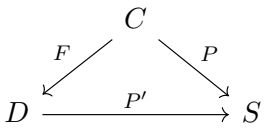
where P and Q are surjective discrete opfibrations. Then R is also a surjective discrete opfibration.

See also Spivak's MathOverflow question [Spi14] for more remarks on why we might want to impose surjectivity.

Definition 3.3. A **minimal schema** for C is a terminal object in $C_{\mathbb{S}}$.

Theorem 3.4. *Let C be such that each x/C has no non-identity automorphisms. Then $C_{\mathbb{S}}$ has a terminal object, given by the construction in the proof of Theorem 2.4.*

Proof. Let $P\colon C \to S$ be the schema constructed in Theorem 2.4, and let $F\colon C \to D$ be a surjective discrete opfibration. Since F is a surjective discrete opfibration, D also has the property that each x/D has no non-identity automorphisms. Further, the isomorphism classes of coslices in D are the same as those in C. Applying Theorem 2.4 to D, we get another surjective discrete opfibration $P'\colon D \to S$. We leave it to the reader to verify that P' is the unique functor making the following diagram commute:



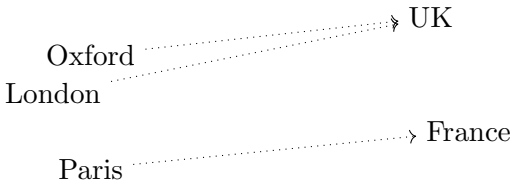
Thus P is a terminal object in $C_{\mathbb{S}}$. □

We have thus proven the minimality of P. This answers Spivak's question [Spi14] for categories satisfying the hypothesis of the theorem.

4 Implications for word embeddings?

We end by speculating about possible implications of our result for word embeddings.

Word embedding algorithms take a corpus of text and embed its words as points in a high dimensional vector space [MCCD13]. A good embedding not only clusters semantically similar words together, but also renders similar *relationships* between words as almost parallel difference vectors between points. For instance, we might obtain an embedding of the form:



(1)

We are interested in the following questions:

1. What kind of mathematical structure best describes the resulting embedding?
2. How do the existing algorithms produce such an embedding?
3. Why do the existing algorithms produce such an embedding?

The results of this paper do *not* answer these questions, at least not in a rigorous sense. However, the outputs of word embeddings and the category of elements of a database share some obvious visual similarities.

In a word embedding algorithm, two words that have similar *relationships* to other words should be clustered close together. Note that this does *not* mean that these two words u, v have the same relationship to a given word w. Rather, if u is related to w in some way, and v is similar to u, then v should be related to some x in the same way, where x is not necessarily w! For instance, 'king' and 'kings' are related in the same way that 'queen' and 'queens' are related, not 'queen' and 'kings'.

Similarly, if two objects u, v of a category C have isomorphic coslice categories $u/C \cong v/C$, it does not mean that $C(u, w) \cong C(v, w)$ for each $w \in C$. Rather, every $u \to w$ has a counterpart $v \to x$ for some x that is not necessarily w. In fact, we could even have two arrows $u \rightrightarrows w$ with the same codomain w corresponding to two arrows $v \to x$ and $v \to y$ where $x \neq y$.

In both word embeddings and isomorphisms of coslice categories, it is the relationships themselves – independent of the objects that they map to or from – that are of primary importance.

Finally, by clustering words with similar relationships ('coslice categories') close together, the resulting word embedding achieves the structure of the category of elements of a database. Similarly, by identifying ('clustering') objects with isomorphic coslice categories, we obtain a schema and a database of which C is the category of elements.

In light of the preceeding discussion, we hope that the following statments will sound like plausible answers to their respective questions:

1. An embedding aims to capture the structure of the *category of elements* of a functor $S \to \mathbf{Set}$.

2. Embedding algorithms work by embedding objects with similar *coslice categories* close together.

3. This works because for categories C satisfying some conditions, there exists a *discrete opfibration* $P\colon C \to S$ such that $Px = Py$ precisely when the coslice categories x/C and y/C are isomorphic.

These statements are necessarily speculative and imprecise. After all, a corpus of text does not have the structure of a category. Nevertheless, we hope that the analogy we have drawn is strong enough to suggest that word embeddings are able to extract approximations of databases from a corpus of text.

References

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013).

[Spi12] David I Spivak, *Functorial data migration*, Information and Computation **217** (2012), 31–51.

[Spi14] David Spivak, *Does there exist a terminal surjective discrete fibration out of C?*, 2014. URL: https://mathoverflow.net/q/157519.