CATEGORICAL DATABASES

SPECIAL GEOMETRY SEMINAR AT U. TEXAS
2012/01/31

DAVID I. SPIVAK

Abstract. There is a fundamental connection between databases and categories. A database, which consists of a "schema" and some "conforming data", is tightly analogous with a category C and a set-valued functor $C \to \mathbf{Set}$. With such a simple connection between seemingly disparate worlds, we can compare the standard operations from category theory with those from database theory. The correspondence is quite strong, and the categorical viewpoint offers a simplification and standardization of ideas such as "schema mapping" (functor), data migration (pullbacks and push-forwards), and others. From a mathematical standpoint, the ideas in this talk will be fairly unsophisticated; the main point is to offer a simple connection between two fields and suggest that categories provide the "right way" to think about databases. I'll also discuss how techniques from homotopy theory relate to the study of information.

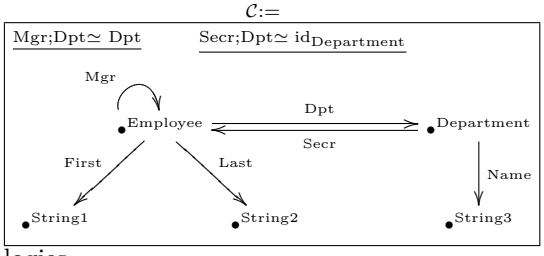
I. Introduction
  A. My goal is to understand what information is.
    1. And to mathematically describe what can be recognized and organized there.
    2. Analogy: if one wanted to understand what symmetry is, one might end up thinking about groups.
  B. I think that math can be helpful.
    1. We put facts together to get new facts.
    2. As Dan Kan says, "information is inherently a combinatorial affair."
    3. What is the combinatorics of information?
    4. Reducing to modus ponens is like reducing athletics to the movement of atoms or all of math to set theory – you may be able to do it, but it's a bad idea.
  C. Shannon's theory is not sufficient.
  D. Where to start – databases.
II. Categorical databases
  A. Databases
    1. Example
      a.

(1)

Employee				
Id	**First**	**Last**	**Mgr**	**Dpt**
101	David	Hilbert	103	q10
102	Bertrand	Russell	102	x02
103	Alan	Turing	103	q10

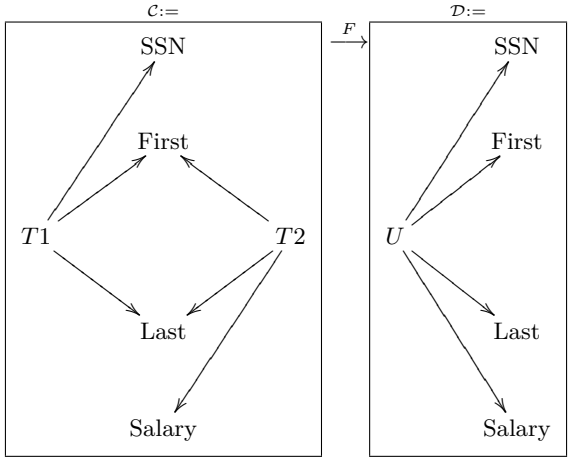
Department		
Id	**Name**	**Secr'y**
q10	Sales	101
x02	Production	102

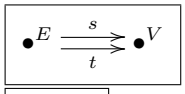
    2. Schemas

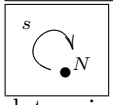
a. 

3. Ontologies
   a. Definition of ontology
   b. Definition of translation
   c. Equivalent to **Cat**.
4. States $\delta\colon \mathcal{C} \to \mathbf{Set}$
   a. $\mathcal{C}-\mathbf{Set}$
   b. What do natural transformations mean?
   c. Note that the schema is intentional not extensional: we allow the extension to change in time, but the intention is fixed.
5. Grothendieck construction $p\colon \int \delta \to \mathcal{C}$

B. Examples
1.

(2) 

2. 

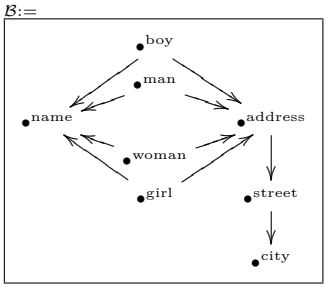
3. 

III. Functorial data migration
A. The migration functors for $F\colon \mathcal{C} \to \mathcal{D}$
1. $F^*, F_*, F_!$
   a. $F_!(\gamma)(d)$ is colimit of $(F \downarrow d) \to \mathcal{C} \xrightarrow{\gamma} \mathbf{Set}$
   b. $F_*(\gamma)(d)$ is limit of $(d \downarrow F) \to \mathcal{C} \xrightarrow{\gamma} \mathbf{Set}$
2. Fact: $F\colon \mathcal{C} \to \mathcal{D}$ is fully faithful iff $F^*F_* \cong \mathrm{id}_{\mathcal{C}-\mathbf{Set}} \cong F^*F_!$

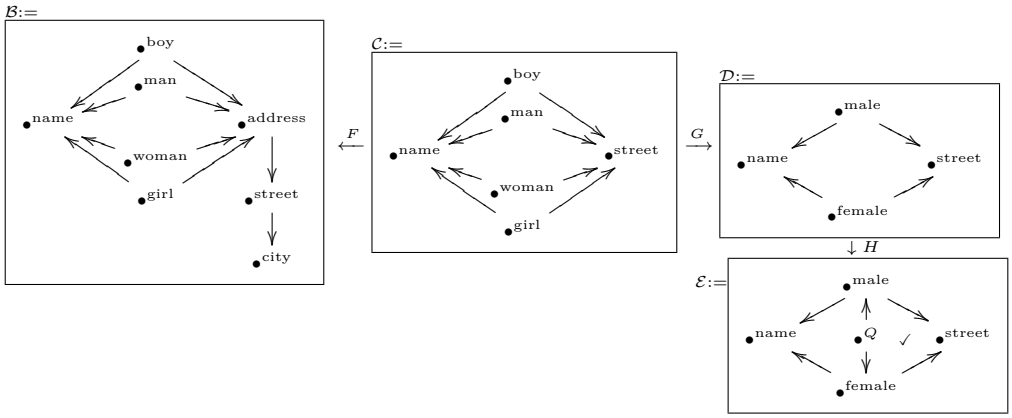
B. Examples
1. Do example 1 for $F^*, F_!, F_*$.

C. Typing
1. What if we want to add typing example 1?
2. **Type**
   a. The type system of Haskell,
      (1) Including objects **Int**, **String**, etc.
      (2) Including morphisms **length**: **String** $\to$ **Int**, etc.
   b. A functor V: **Type** $\to$ **Set**
3. Now suppose we need some fragment of it to give types to $\mathcal{C}$.
   a. We begin with **Type** $\xleftarrow{F}$ $\mathcal{B}$ $\xrightarrow{G}$ $\mathcal{C}$
   b. Now look at $\mathcal{C}-\mathbf{Set}_{/G_*F^*V}$
   c. Still a topos. Changing types gives essential geometric morphisms (all three migration functors).

IV. Queries
A. Joins, unions, etc.
1. Beginning with

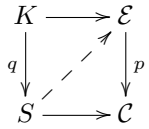


2. We want males and females who live on the same street.

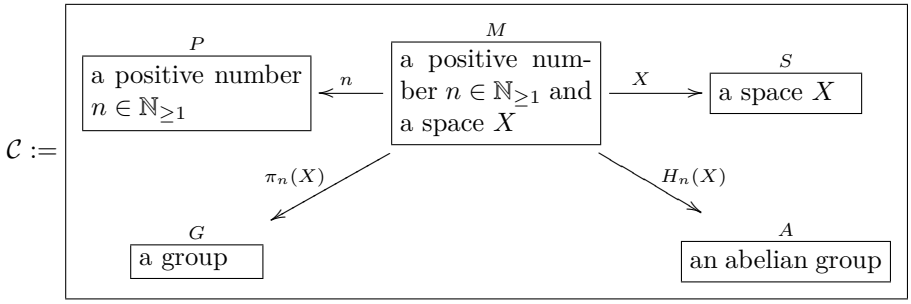


B. Graph patterns via "Lifting diagrams"
1. A boy and a girl live on the same street; the boy's name is John and the girl's name is Sue. Find their addresses.
2. Strategy:
   a. Given data p: $\mathcal{E} \to \mathcal{C}$, a query on p is given by two things: the shape relating what we know and what we want to know, and data we know.

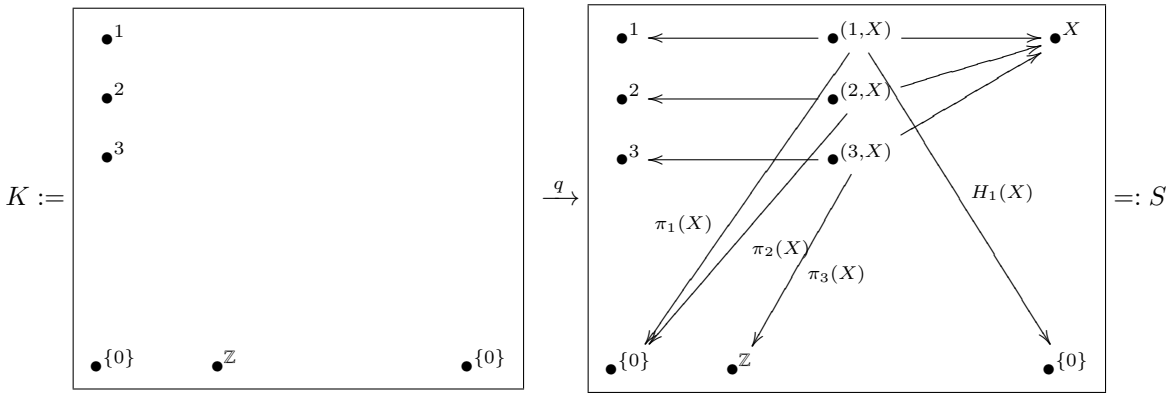
b. Write this as the commutative diagram; we're looking for lifts



3. Instead of our John and Sue example, let's do a more mathematical example:
   a. Tell me all known spaces X with $\pi_1(X) = \pi_2(X) = 0, \pi_3(X) = \mathbb{Z}$ and $H_1(X) = 0$.
   b. Suppose we have data $p\colon \mathcal{E} \to \mathcal{C}$ on the olog



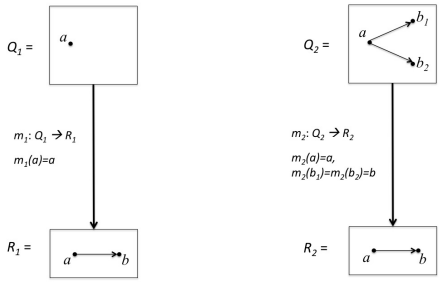
   c. Suppose we have some data input, $p\colon \mathcal{E} \to \mathcal{C}$
   d. Lift



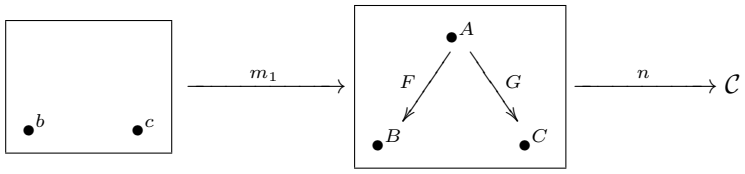
4. Integer sequence database – similar idea, but less sophisticated.

C. Constraints

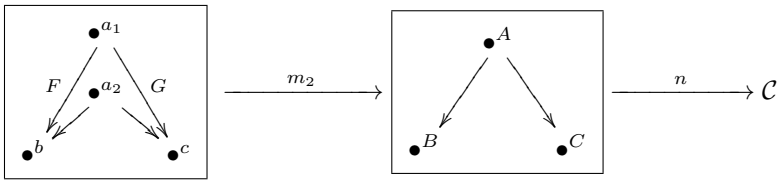
1. Given a functor $\pi\colon \mathcal{D} \to \mathcal{C}$, it is a data fibration if it lifts all constraints



2. You can declare many constraints using local lifting conditions: surjection, injection, fiber product, empty table, card=1, etc.
   a. Form: $\{Q \xrightarrow{m_i} R \xrightarrow{n_i} \mathcal{C}\}$.
   b. Example: A is the product of B and C.
      (1) The existence constraint is



      (2) The uniqueness constraint is



3. An adaptation of Garner's small object argument can be used to universally correct any database state to one satisfying the constraints.

V. Applications
   A. Materiomics
      1. "Solving the olog" (system of equations)
      2. Better articulate scientific definitions and conclusions to make it more rigorous.
      3. The categorical model just works – people find it easy to use.
   B. Network of interaction
      1. A simplicial complex (or semi-simplicial set)
      2. To each simplex assign a category (the world-view)
      3. A functor from higher-dimensional "common ground world-view" category to each sub-simplex's world-view.
      4. We already saw this in the typing schema $\mathbf{Type} \leftarrow \mathcal{B} \rightarrow \mathcal{C}$.
      5. Web of science / math could be useful – navigate semantically.
   C. Homotopical view into data
      1. Let $I \rightarrow S$ be a data bundle.
      2. Taking the nerve, we get a topological space $N(I)$.
      3. If S is simplicial indexing category, we get the right answer. Can this be useful in understanding data?