TOPOLOGY AND INFORMATION
UNIVERSITY OF CHICAGO 2010/05/11

DAVID I. SPIVAK

Abstract. Piecing together information on a topic may sound roughly like a matter of taking colimits, but in what kind of category? In an effort to understand what information is, I began by trying to digest databases using category theory. The result was a surprising connection to topology: a database is a pair $(X, \mathcal{O}_X)$ where X is a simplicial set and $\mathcal{O}_X$ is a sheaf on it; a morphism of databases follows the typical pattern for a morphism of such "structured spaces."

The above formulation is nice in that the categorical and topological properties of these structured spaces have information-theoretic meaning. For example colimits and limits in this category yield the typical database manipulations (adding new information or merging databases). Curves through the simplicial set X correspond to queries of the database $(X, \mathcal{O}_X)$ and these curves form an interesting 2-category. In this talk I'll define the category of databases, explain the above ideas, and show how the above theory generalizes that of multi-categories (i.e. colored operads).

Thank Vigleik Angeltveit.

I. Introduction.
   A. What is information?
      1. Putting together facts in a legal and coherent way: new facts from old.
      2. Dan Kan: "Information is inherently a combinatorial affair."
      3. Hilbert: "This formula game is carried out according to certain definite rules, in which the technique of our thinking is expressed. [...] The fundamental idea of my proof theory is none other than to describe the activity of our understanding, to make a protocol of the rules according to which our thinking actually proceeds" (Hilbert, 1928, 475).
   B. Where is it found?
      1. Internet: Semantic Web
      2. Mathematics itself.
      3. Fundamental: databases.
         a. Differential equations are to physics as databases are to information.
   C. What is a database?
      1. Linked tables
      2. Draw ER-diagram
      3. What are the morphisms?
      4. Join of two tables: colimit/limit mystery.

II. The category of tables.

A. Definition

1. Fix $\pi\colon \mathbf{U} \to \mathbf{DT}$.

2. Schema: $\sigma\colon C \to \mathbf{DT}$.

3. Records: $\Gamma(C, \sigma) = \{f\colon C \to \mathbf{U} \mid \pi \circ f = \sigma\}$.

4. Table: $\delta\colon R \to \Gamma(C, \sigma)$.

a. A set of vector fields.

5. Morphisms of tables:

a. $f\colon R_1 \to R_2$,

b. $g\colon C_2 \to C_1$ such that $\sigma_1 \circ g = \sigma_2$ and such that

c. the diagram

$$\begin{tikzcd}
R_1 \arrow[r, "\delta_1"] \arrow[d, "f"'] & \Gamma(C_1, \sigma_1) \arrow[d, "\Gamma(g)"] \\
R_2 \arrow[r, "\delta_2"'] & \Gamma(C_2, \sigma_2)
\end{tikzcd}$$

commutes

6. Simplest: Projections. $C_2 \subseteq C_1$.

B. Limits and colimits

1. Unions and insertions

2. Joins and selects

a. Given a table $\delta\colon R \to \Gamma(C)$ and something to select: $C' \subseteq C'$ and $R' \to \Gamma(C')$

b. The limit of

$$\begin{tikzcd}
R \arrow[r, "\delta"] \arrow[d] & \Gamma(C) \arrow[d] \\
\Gamma(C') \arrow[r] & \Gamma(C') \\
R' \arrow[r] \arrow[u] & \Gamma(C') \arrow[u]
\end{tikzcd}$$

is called "selecting from δ."

III. The category of databases.

A. How tables relate in a database

1. ER-diagram picture.

2. Different tables may share common columns.

3. Information can then flow from one table to another.

4. Some tables have incomplete information.

B. Schemas.

1. Semi-simplicial set X with vertices labeled in $\mathbf{DT}$.

2. Map $\sigma\colon X \to \check{C}(\mathbf{DT})$

3. Universal sheaf: $\mathcal{U}_X\colon \mathbf{Sub}(X)^{\mathrm{op}} \to \mathbf{Sets}$

a. $\mathcal{U}_X(V) = \{f\colon V \to \check{C}(\mathbf{U}) \mid \pi \circ f = \sigma\}$

C. Databases

1. Sheaf $\mathcal{O}_X\colon \mathbf{Sub}(X)^{\mathrm{op}} \to \mathbf{Sets}$

2. Together with map $\delta\colon \mathcal{O}_X \to \mathcal{U}_X$.

3. Total: write $(X, \mathcal{O}_X)$ to denote $(\sigma\colon X \to \check{C}(\mathbf{DT}), \delta\colon \mathcal{O}_X \to \mathcal{U}_X)$.
4. Morphism $(f, f^\sharp)\colon (X, \mathcal{O}_X) \to (Y, \mathcal{O}_Y)$.
5. Note that the category of databases on a schema is a topos, and a map is a geometric morphism.

IV. Manipulations, queries, and constraint problems.

A. Everything done to tables, can be done locally in a database.

1. Insert, union
2. join, select, delete

B. Mix and match tables for reasoning. (Minority report).

1. Drag and drop.
2. Click to view

C. Global sections correspond to "solutions to constraint problems."

1. $(a + b = c) \amalg (bc = d) \amalg (a = d)$.
2. Fermat's last theorem example.

D. Queries are curves

1. A curve through X corresponds to a zig-zag in $Gr(X)$.
   a. Note that $\mathbf{Tables} \xrightarrow{\Sigma} (\mathbf{Fin} \downarrow \mathbf{DT})^{\mathrm{op}}$ is a split fibration and a split op-fibration.
2. Down to earth: given zigzag $\sigma_1 \to \sigma_2 \leftarrow \sigma_3 \to \sigma_4 \leftarrow \sigma_5$.
   a.

$$\begin{tikzcd}
K_1 \arrow[r, equal] \arrow[d] & \overset{K_2:=}{K_1} \arrow[d] & \overset{K_3:=}{(K_2 \times_{R_2} R_3)} \arrow[l] \arrow[r, equal] \arrow[d] \arrow[dl, phantom, very near start, "\llcorner"] & \overset{K_4:=}{K_3} \arrow[d] & \overset{K_5:=}{(K_4 \times_{R_4} R_5)} \arrow[l] \arrow[d] \arrow[dl, phantom, very near start, "\llcorner"] \\
R_1 \arrow[r] \arrow[d, "\delta_1"] & R_2 \arrow[d, "\delta_2"] & R_3 \arrow[l] \arrow[r] \arrow[d, "\delta_3"] & R_4 \arrow[d, "\delta_4"] & R_5 \arrow[l] \arrow[d, "\delta_5"] \\
\Gamma(\sigma_1) \arrow[r] & \Gamma(\sigma_2) & \Gamma(\sigma_3) \arrow[l] \arrow[r] & \Gamma(\sigma_4) & \Gamma(\sigma_5) \arrow[l]
\end{tikzcd}$$

3. The point is, I have a map $K_5 \to \Gamma(\sigma_5) \times \Gamma(\sigma_1)$ which shows where everything went in the query.
4. May want to consider some kind of 2-category – unsure.

V. Operational tables and multi-categories.

A. In practice most tables in a database have a single "primary key column."

1. This is because they need to protect themselves against the ignorance of no category theory.

B. Given a table T, with non-key columns $c_1, \ldots c_n$, attach n other tables $T_1, \ldots T_n$ via their primary key.

1. Typically, you often "join" these together to get a new table (DRAW IT).
2. We can call this composition.

C. Conversion to and from multi-categories.

1. Given a set of tables (each with primary keys), close it under composition.
2. This gives a multi-category.
3. Given a small multi-category M,
   a. Data types:

(1) Make $\mathbf{DT} = \mathbf{Ob}(M)$,

(2)

$$\mathcal{U} = \coprod_{\substack{n \geq 1 \\ A_1, \ldots, A_n, B}} \mathbf{Hom}(A_1, \ldots, A_n; B),$$

(3) π sends $f \colon (A) \to B$ to $B \in \mathbf{DT}$.

b. Tables:

(1) For each $f \colon (X_1, \ldots, X_n) \to Y$ in M

(2) a table T_f with columns $X_1, \ldots, X_n, Y$ with B the primary key

(3) rows: $(g_1, \ldots, g_n, h)$ such that $g_i \in \coprod_{(A)} \mathbf{Hom}((A), X_i)$ and $h = f \circ (g_1, \ldots, g_n)$.

4. This is an adjunction (multi-categories is reflexive subcategory of "compositional directed table systems")
5. We can compose any finite set of tables by looking at outer vertices.

a. For example, composition of fermat's last database is $\{(a, b, c) | \exists n \geq 3, a^n + b^n = c^n\}$.

b. It's like a category except with more operations.

c. Final image: simplices with sheaves of sets – meaningful tiles.