CATEGORICAL INFORMATION THEORY

TOPOLOGY SEMINAR AT JHU 2011/11/21

DAVID I. SPIVAK

ABSTRACT. Classical information theory, as developed by Claude Shannon in 1948, studies how to optimize the quantity of data that can be transmitted across a noisy channel, but it ignores what this information "means". It is clear that information is intended to mean or signify something – this is its only purpose – but how can such a thing as meaning be formalized? In this talk I'll discuss how category theory may be useful in this endeavor. Once the basics (which are quite straightforward) are clear, I'll discuss some interesting applications to computer science and materials engineering (!), and time permitting look into the vast array of open and interesting avenues for future work.

Thanks to John Lind and Jack Morava for the invitation.

OUTLINE OF TALK

I. Introduction
  A. What is information?
    1. Examples
    2. Information vs. data
    3. Not absolute but relational
    4. Intention vs. extension
  B. Historical context
    1. Frege – Begriffsschrift
    2. 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).
    3. Dan Kan: "Information is inherently a combinatorial affair." Facts are fit together in appropriate ways to create new facts.
    4. Databases
  C. Outline of talk
    1. Information via category theory
      a. Main idea: databases are custom categories
      b. Aside from speed considerations, DB theory is category theory.
      c. Formulations beloved by topologists show up in databases
    2. Functorial data migration
    3. Queries
    4. Communication networks
    5. Applications

II. Information via category theory
  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.

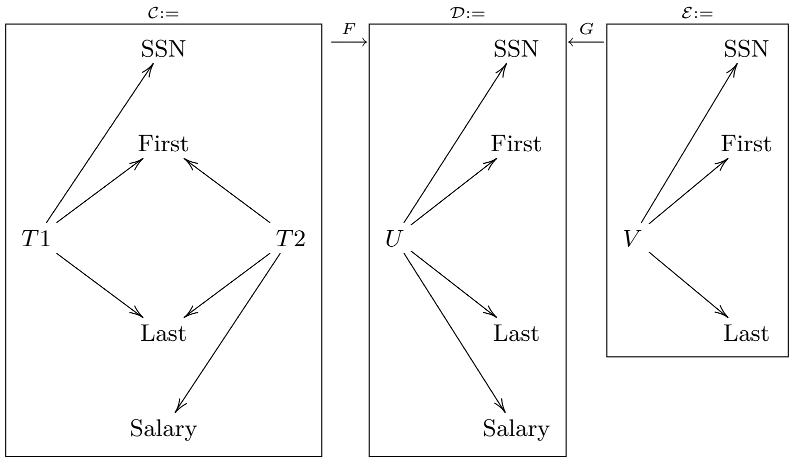
(2)

$$\mathcal{C} := \boxed{\begin{tikzcd} \bullet \text{Employee} \arrow[loop above, "\text{Mgr}"] \arrow[r, shift left, "\text{Dpt}"] \arrow[d, bend right, "\text{First}"'] \arrow[d, bend left, "\text{Last}"] & \bullet \text{Department} \arrow[l, shift left, "\text{Secr'y}"] \arrow[dl, bend left, "\text{Name}"] \\ \bullet \text{String} & \end{tikzcd}}$$

3. States $\delta\colon \mathcal{C} \to \mathbf{Set}$
   a. Note that the schema is intentional not extensional: we allow the extension to change in time, but the intention is fixed.
4. Grothendieck construction $p\colon \int \delta \to \mathcal{C}$

B. Examples
1.

(3)



2. $$\boxed{\begin{tikzcd} \bullet E \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & \bullet V \end{tikzcd}}$$
3. $$\boxed{\begin{tikzcd} X \arrow[loop left, "s"] \end{tikzcd}}$$

III. Functorial data migration
A. The migration functors for $F\colon \mathcal{C} \to \mathcal{D}$
1. $F^*, F_*, F_!$

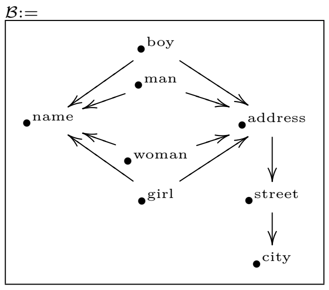
a. $F_!(\gamma)(d)$ is colimit of $(F\downarrow d)\to\mathcal{C}\xrightarrow{\gamma}\mathbf{Set}$
b. $F_*(\gamma)(d)$ is limit of $(d\downarrow F)\to\mathcal{C}\xrightarrow{\gamma}\mathbf{Set}$

2. Fact: $F\colon\mathcal{C}\to\mathcal{D}$ is fully faithful iff $F^*F_*\cong\mathrm{id}_{\mathcal{C}-\mathbf{Set}}\cong F^*F_!$

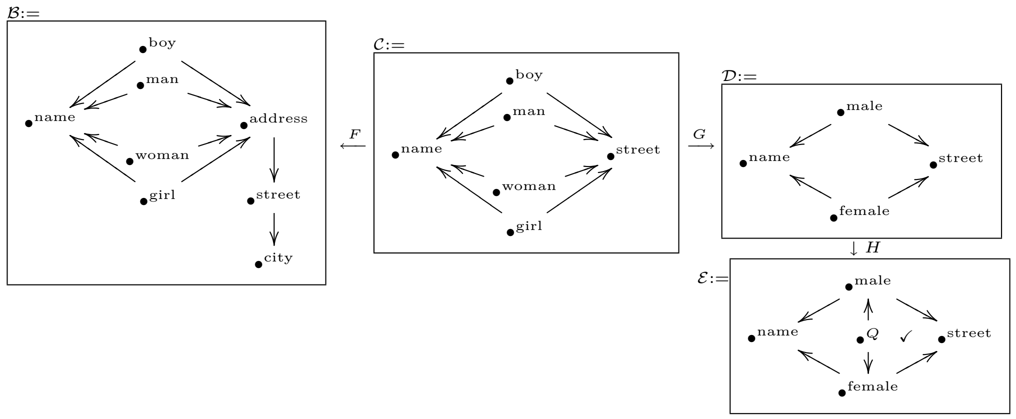
B. Examples
1. Do example 1 for $F_!, F_*, G_!, G_*$.

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 $V\colon\mathbf{Type}\to\mathbf{Set}$
3. Now suppose we need some fragment of it to give types to $\mathcal{C}$.
a. We begin with $\mathbf{Type}\xleftarrow{F}\mathcal{B}\xrightarrow{G}\mathcal{C}$
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 $p\colon \mathcal{E} \to \mathcal{C}$, 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

$$\begin{tikzcd}
K \arrow[r] \arrow[d, "q"'] & \mathcal{E} \arrow[d, "p"] \\
S \arrow[r] \arrow[ur, dashed] & \mathcal{C}
\end{tikzcd}$$

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

$$\mathcal{C} := \boxed{\begin{tikzcd}
\overset{P}{\boxed{\text{a positive number } n \in \mathbb{N}_{\geq 1}}} & \overset{M}{\boxed{\text{a positive number } n \in \mathbb{N}_{\geq 1} \text{ and a space } X}} \arrow[l, "n"'] \arrow[r, "X"] \arrow[dl, "\pi_n(X)"'] \arrow[dr, "H_n(X)"] & \overset{S}{\boxed{\text{a space } X}} \\
\overset{G}{\boxed{\text{a group}}} & & \overset{A}{\boxed{\text{an abelian group}}}
\end{tikzcd}}$$

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

$$K := \boxed{\begin{tikzcd}
\bullet^{1} & & \\
\bullet^{2} & & \\
\bullet^{3} & & \\
\bullet^{\{0\}} & \bullet^{\mathbb{Z}} & \bullet^{\{0\}}
\end{tikzcd}} \xrightarrow{q} \boxed{\begin{tikzcd}
\bullet^{1} & \bullet^{(1,X)} \arrow[l] \arrow[r] \arrow[ddd, "\pi_1(X)"'] \arrow[dddr, "H_1(X)"] & \bullet^{X} \\
\bullet^{2} & \bullet^{(2,X)} \arrow[l] \arrow[ur] \arrow[dd, "\pi_2(X)"] & \\
\bullet^{3} & \bullet^{(3,X)} \arrow[l] \arrow[uur] \arrow[d, "\pi_3(X)"] & \\
\bullet^{\{0\}} & \bullet^{\mathbb{Z}} & \bullet^{\{0\}}
\end{tikzcd}} =: S$$

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

V. Communication networks...
   A. 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
   B. Communication protocol
      1. a. A *piece of knowledge of A* is a pair (I, f) where I is a finite category and $f\colon I \to K_A$ is a functor; the category I is called the *formation* of the knowledge.

b. Given a piece of knowledge $f\colon I' \to K_A$ of A, a diagram

$$\begin{tikzcd} I \arrow[r] \arrow[d] & K_{AB} \arrow[r] \arrow[d] & K_B \\ I' \arrow[r] & K_A & \end{tikzcd} \tag{4}$$

is called a *communication attempt from A to B*;

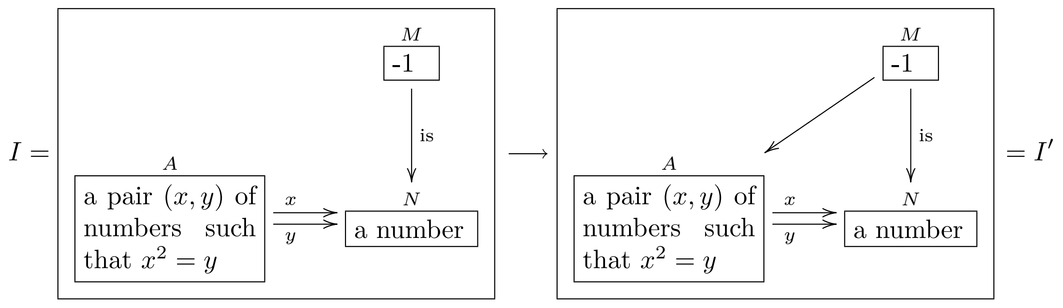
c. An *interpretation by B of $\bar{I}$* is a diagram

$$\begin{tikzcd} I \arrow[r] \arrow[d] & K_{AB} \arrow[r] \arrow[d] & K_B \\ I' \arrow[r] \arrow[dr] & K'_{AB} \arrow[ur] \arrow[d] \arrow[ul, phantom, very near start, "\ulcorner"] & \\ & K_A & \end{tikzcd}$$

d. A *learning by B of $\bar{I}$* is a diagram

$$\begin{tikzcd} I \arrow[r] \arrow[d] & K_{AB} \arrow[r] \arrow[d] & K_B \arrow[d] \\ I' \arrow[r] \arrow[dr] & K'_{AB} \arrow[r] \arrow[d] \arrow[ul, phantom, very near start, "\ulcorner"] & K'_B \arrow[ul, phantom, very near start, "\ulcorner"] \\ & K_A & \end{tikzcd}$$

2. Example:


$I =$
M
-1
is
A
a pair (x, y) of numbers such that $x^2 = y$
x
y
N
a number
$\longrightarrow$
M
-1
is
A
a pair (x, y) of numbers such that $x^2 = y$
x
y
N
a number
$= I'$

VI. Applications

A. Materiomics

1. "Solving the olog" (system of equations)
2. Bio-inspired materials – make it rigorous.

B. Web of science?

1. If papers came with ologs specifying their content, graduate students could add data as they work out examples.