Abstract Algebras:
Category Theory for Biomedicine

Eric Neumann
PSB 2016

Material Used

• Category Theory for Scientists, Spivak 2013
• Categorical Databases, Spivak 2012
• Categories for the Working Mathematician, Mac Lane 1998
• Conceptual Mathematics: A First Introduction to Categories
  Lawvere and Schanuel 2009
• Sheaves in Geometry and Logic: A First Introduction to Topos
  Theory, MacLane and Moerdijk, 1994
Category Theory (CT) Basics

• All Categories consist of:
  • Objects (more like what we think of as classes)
  • Morphisms (functions, maps, arrows between objects)
  • Morphisms can be composed in a sensible way: \( g \circ f: x \mapsto g(f(x)) \)
  • \( \text{Hom}_C(c, d) \) = the set of all functions/maps between \( c \) and \( d \)
• Categories are often constructed in the simplest form possible
  • Set, Graph, Grp, Mon, Vect, Top, etc ... even Cat
  • Generality is the name of the game!

Category Theory (CT) Basics

• Functors \( F \) map Categories (C) to Categories (D): \( F: C \rightarrow D \)
  • \( F: \text{Obj}(C) \rightarrow \text{Obj}(D) \), e.g., \( c \mapsto F(c) = d \)
  • \( F: \text{Morphism}(C) \rightarrow \text{Morphism}(D) \), e.g., \( f \mapsto F(f) \)
• For example, \( \text{GS: Graph} \rightarrow \text{Set} \) maps Graph objects and arrows to sets of objects and sets of functions in Set:
  Think Biochemical Networks to Genes and Reactions!
• There’s even morphisms between Functors called Natural Transformations... turtles all the way up!
Category Theory (CT) Basics

Where can one apply CT?

- Sets and their functions (general rules for arithmetic, data)
- All areas of Math and Groups, incl. Topology (String Theory)
- All Concepts that can be formally represented: Schemas and Ontologies (Functors ~ logic, Inference, concept generation)
- All Databases (Functors ~ data migration, queries)
- All Scientific Models: ontological, dynamic, analytical

Power of CT is that theorems in one parent CAT can be applied to all proper subsumed categories!

Just be sure to follow the formal rules!
Practical Categories

Some Science...

CT Universal Properties

- Products, Pullbacks
- Limits
- Equalizers
- Sums, Pushouts
- Commutation
CT Universal Properties

- Products, Pullbacks
- Limits
- Equalizers
- Sums, Pushouts
- Commutation

Functions in CT

Given function/map f: X -> Y ≡ x → f(x)

Exponentiation: f ~ Y^X, means each f is identified by name: Y^X ~ "f"
- "How many ways to maps from X to Y?"
- Ex 1: For an input set X of size |X| mapping to any subset of X (Y ~ membership is binary, so |Y|=2), the number of possible subsets is |Y|^{|X|} (i.e., each x∈X is either included or not), hence there are just as many maps.
- Ex 2: A function with 3 boolean args must have values for 2^3=8 different inputs
- Ex 3: For seq’s X of a similar length and Y, the 4 bases one can choose from, then 4^{|X|} is the number of possible sequences for a string of length |X|

Evaluation: {f(X) = Y} ~ Y^X → ev: X × Y^X → Y

Currying: X × Y →_f Z ~ X →_g Z , where g_y = f(y,_) ~ (Z^Y)^X = (Z^{X×Y})

Exponentials objectify functions and maps!
Path Commutation

\[ f = \text{price} \cdot \text{kind} \]

\[ \sum_{k} \text{price}(k) \cdot (\text{size of the stack of kind over } k) \]

Protein

\[ e = \text{ts} \cdot \text{tl} \]

# Protein from Gene = # RNA from Gene + # Protein from RNA

Prot express rate per Gene = RNA transcribe rate per Gene + Prot transl rate amount per RNA

Protein Seq complexity for Gene = Gene Seq complexity + RNA Seq complexity

Protein->Gene coord = Prot->RNA coord + Gene->RNA coord

G-Sheaf

Variant Sheaf

Structural dependence (information)

Sequence Topology
Some obvious *Products*

- Gene-expr * Time
- Prot-expr * Tissue
- P-Interacts * Gene-expr
- P-Interacts * Prot-expr
- Gene-expr * G-Interacts
- Gene-expr * G-Interacts * Time
- Gene-expr * G-Interacts * Prot-expr
- Gene-expr * G-Interacts * Prot-expr * Tissue * Time
- Prot Structure * Mut Seq-space
- Prot Structure * P-Interacts
- Prot Structure * P-Interacts * Mut Seq-space
- Prot-Act * Prot-expr * Mut Seq-space
- Cell Processes * Prot-Act * Prot-expr * Mut Seq-space
- Onco Processes * Prot-Act * Mut Seq-space * Therapy * Outcomes
OLOGs

- “Ontology Log” – Spivak & Kent, PLOS 2012
- Dynamic Data Schema
  - compose new concepts via U-operations
- Graph formalism w/ path constraints
  - As powerful as Ontology + Rules
- Great for whiteboarding new semantics

Product as a Pullback

\[ D = B \times C \]

- a Signaling Oncogene
- a Signaling Gene
- a Oncogene
- a Gene

Pullback
Product as a Pullback

\[ \text{PaFents with a Molecular Aberration-Disease Association} \]

\[ \text{Patients with a Molecular Aberration} \]

\[ \text{A Disease} \]

\[ \text{B} \]

\[ \text{C} \]

\[ \text{D} = B \times C \]

OLOGs
– rules as programs

\[ f(n) = n! \]

\[ s; p = \text{id}_A \]

\[ s; q = d; f \]

\[ i_0; f = \omega \]

\[ i_1; f = s; m \]

\[ B = A \times D \]

A positive natural number

A pair \((p, q)\) where \(p\) is a positive natural number and \(q\) is a natural number

A natural number

A natural number

Spivak and Kent, PLOS, 2012
Endomaps

\[ C := \begin{array}{c}
\alpha \circ \bullet \\
\end{array} \]

\[ x' = \alpha \circ \alpha \circ x = \alpha(\alpha(x)) \]

Idempotent
\[ x = \beta \circ \beta \circ x \]

*from Lawvere and Schanuel, 2009*
Endomap Products

e.g. pedigree & clans

Graph Products

ex: Mitochondrial Genomes
Graph Definition

\[ C := \begin{array}{c}
\bullet & \stackrel{\text{src}}{\rightarrow} & \bullet \\
A & \stackrel{\text{tgt}}{\rightarrow} & V
\end{array} \]

\[ \sim \]

\[ \begin{align*}
V & \quad A & \quad V \\
<\text{Subj}> & <\text{Prop}> & <\text{Obj}>
\end{align*} \]

It’s the basis of a Triple!

Relation to Shapes

Count commutation

[Diagram of three shapes connected by arrows and equations]

\[ \forall z \{ \Sigma_{x \in X} (h(x)=z) = \Sigma_{y \in Y} (g(y)=z \cdot \Sigma_{x \in X} f(x)=y) \} \]

: h cat:comm_eq (:f :g)

\[ h \cdot s = 1_z \quad : h \text{ cat:right_inv :s} \]

This can also be a data constraint!!
... even deeper Logic Applications

Theorem 5 (The Recursion Theorem) Let \( h : \mathbb{N} \to \mathbb{N} \) be a total computable function. There exists an \( n_0 \in \mathbb{N} \) such that
\[
\phi_{\langle n_0 \rangle} = \phi_{\langle n_0 \rangle}.
\]
Proof. Let \( \mathcal{F} \) be the set of unary computable functions. Consider \( f : \mathbb{N} \times \mathbb{N} \to \mathcal{F} \) be defined as \( f(m, n) = \phi_{\langle n_0 \rangle} \). If \( \phi_{\langle n_0 \rangle} \) is undefined, then \( f(m, n) \) is also undefined. Letting the operator \( \phi_{\langle n_0 \rangle} : \mathcal{F} \to \mathcal{F} \) be defined as \( \phi_{\langle n_0 \rangle} \phi_{\langle n_0 \rangle} = \phi_{\langle n_0 \rangle} \).

Yoneda lemma

The Yoneda lemma allows the embedding of any category into a category of functors defined on that category. It suggests that instead of studying the (small) category \( \mathcal{C} \), one should study the category of all functors of \( \mathcal{C} \) into \( \mathbf{Set} \) (the category of sets with functions as morphisms). \( \mathbf{Set} \) is a category we understand, and a functor of \( \mathcal{C} \) into \( \mathbf{Set} \) can be seen as a "representation" of \( \mathcal{C} \) in terms of known structures. The original category \( \mathcal{C} \) is contained in this functor category, but new objects appear in the functor category, which were absent and "hidden" in \( \mathcal{C} \).

For a functor \( F \) and for CAT \( \mathcal{C} \) with objects A, B, C and morphisms f,g,h...

All \( \phi_{\langle n \rangle}(x) \) must have a fixed point for some \( t \) specifically \( \phi_{\langle n \rangle}(x) = \phi_{\langle n \rangle}(t) \) otherwise FALSE: e.g., \( h(f) = \text{neg}(f) \), Gödel's Incompleteness Theorem

Yanofsky 2003
Yoneda lemma applied to databases

For instances (functor) I and for Schema category objects (tables) A, B, C...

\[ h^A = \text{Hom}(A, _) \quad \text{Nat}(h^A, I) = I(A) \]

\[
\begin{align*}
\text{Nat}(h^A, I)(_) &\rightarrow \text{Hom}(h^A(_), I(\_)) \\
h^A(B) &= \text{Hom}(A, B) \quad \text{Hom}_I(h^A(B), I(B)) = I_B(A) \\
h^A(C) &= \text{Hom}(A, C) \quad \text{Hom}_I(h^A(C), I(C)) = I_C(A)
\end{align*}
\]

CT application areas

1. Valid Science concept (domain) representations
   - Functional maps, dependencies, limits/products
2. Rules for database design and use
   - Data integrity, Schema migration, Ontology alignment, query forms
3. Rules for inference and knowledge discovery
   - Conditions for productization, exponentiation, Discovery Functors, Tool Wrappers (monads)
4. A formal bridge between CT and LD could help accelerate the development of tools and applications for many areas of Data Sciences
Concluding remarks

• Should we begin a regular, semi-formal discussion group around CT and LD?
• PSB2016 Workshop on Abstract Algebra and Topology