# Hierarchical Ontology and Knowledge Representation

Noah Chrein

University of Maryland

October 26, 2019



#### In philosophy, ontology is "a collection of things that exist"



In philosophy, ontology is "a collection of things that exist"

- Set
- Graph
- Category

In philosophy, ontology is "a collection of things that exist"

- Set
- Graph
- Category
- Even this list!



### David Spivak's idea of an Ontological Log



### David Spivak's idea of an Ontological Log Categories as a database to house arbitrary information





### David Spivak's idea of an Ontological Log Categories as a database to house arbitrary information



Picture taken from [OLOG]

э



#### Goal

The goal is to define an rigorous and expressive notion of **Ontology** 

This includes:

A means of organizing data



▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00



#### Goal

The goal is to define an rigorous and expressive notion of **Ontology** 

This includes:

- A means of organizing data
- A means of elaborating on data



#### Goal

The goal is to define an rigorous and expressive notion of **Ontology** 

### This includes:

- A means of organizing data
- A means of elaborating on data
- A means of recovering data from elaborations



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

In philosophy, Ontology is "a collection of things that exist"



▲□▶ ▲□▶ ▲ □▶ ▲ □▶ □ のへぐ



Sets

Graphs



Sets

Graphs

Simplicial Sets



Sets

Graphs

Simplicial Sets

**Globular Sets** 



▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三 のへぐ

Sets

Graphs

Simplicial Sets

**Globular Sets** 

Which are all functors  $\Delta^{op} \to \mathfrak{Set}$ 

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Sets  $* \to \mathfrak{Set}$ Graphs Simplicial Sets Globular Sets Which are all functors  $\Delta^{op} \to \mathfrak{Set}$ 

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

Sets	$*  ightarrow \mathfrak{Set}$
Graphs	$G^{op}  ightarrow \mathfrak{Set}$
Simplicial Sets	
Globular Sets	
Which are all functors $\Delta^{op}$ —	→ Set

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

Sets	$*  ightarrow \mathfrak{Set}$
Graphs	$G^{op}  ightarrow \mathfrak{Set}$
Simplicial Sets	$\Delta^{op}  o \mathfrak{Set}$
Globular Sets	
Which are all functors $\Delta^{op}$ .	ightarrow Set

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

Sets	$*  ightarrow \mathfrak{Set}$
Graphs	$G^{op}  ightarrow \mathfrak{Set}$
Simplicial Sets	$\Delta^{op}  o \mathfrak{Set}$
Globular Sets	$\mathbb{G}^{op}  ightarrow \mathfrak{Set}$

Which are all functors  $\Delta^{op} \to \mathfrak{Set}$ 

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

Sets	$*  ightarrow \mathfrak{Set}$
Graphs	${\mathcal G}^{op}  o \mathfrak{Set}$
Simplicial Sets	$\Delta^{op}  o \mathfrak{Set}$
Globular Sets	$\mathbb{G}^{op}  ightarrow \mathfrak{Set}$
Which are all functors	$\Delta^{op}  ightarrow \mathfrak{Set}$

#### Goal 1

An Ontology should organize data

### **Basic Ontology**

- A basic ontology is a functor  $\Sigma:\Delta^{\textit{op}}\to \mathfrak{C}.$  Where
  - $\blacksquare$   $\Delta$  is a small category
  - C is an arbitrary (possibly higher) category

### **Basic Ontology**

- A basic ontology is a functor  $\Sigma:\Delta^{\textit{op}}\to \mathfrak{C}.$  Where
  - $\blacksquare$   $\Delta$  is a small category
  - C is an arbitrary (possibly higher) category

Only organizes data, makes organizational statements

### Basic Ontology

- A basic ontology is a functor  $\Sigma:\Delta^{\textit{op}}\to \mathfrak{C}.$  Where
  - $\blacksquare$   $\Delta$  is a small category
  - C is an arbitrary (possibly higher) category

Only organizes data, makes organizational statements

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

•  $\Delta$  is the "Organizational Shape"

### **Basic Ontology**

A basic ontology is a functor  $\Sigma : \Delta^{op} \to \mathfrak{C}$ . Where

- Δ is a small category
- C is an arbitrary (possibly higher) category

Only organizes data, makes organizational statements

- Δ is the "Organizational Shape"
- C is the "Propositional Category"

(For example  $\sigma \in \Sigma[2]$  is a statement when  $\mathfrak{C} = \mathfrak{Set}$ )

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

#### **Basic Ontology**

A basic ontology is a functor  $\Sigma : \Delta^{op} \to \mathfrak{C}$ . Where

- Δ is a small category
- C is an arbitrary (possibly higher) category

Only organizes data, makes organizational statements

- Δ is the "Organizational Shape"
- C is the "Propositional Category"

(For example  $\sigma \in \Sigma[2]$  is a statement when  $\mathfrak{C} = \mathfrak{Set}$ ) at the end of the day, the "data" is just an element of a set

### Basic Ontology

A basic ontology is a functor  $\Sigma : \Delta^{op} \to \mathfrak{C}$ . Where

- Δ is a small category
- C is an arbitrary (possibly higher) category

Only organizes data, makes organizational statements

- Δ is the "Organizational Shape"
- C is the "Propositional Category"

(For example  $\sigma \in \Sigma[2]$  is a statement when  $\mathfrak{C} = \mathfrak{Set}$ ) at the end of the day, the "data" is just an element of a set

Hence the name "basic": just a framework

### For mathematical concepts, composition is usually well defined

(ロ)、(型)、(E)、(E)、 E) の(()

For mathematical concepts, composition is usually well defined But for more abstract concepts:



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

we might be at a loss

For mathematical concepts, composition is usually well defined But for more abstract concepts:

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで



If we give a "reasonable" composition:



If we give a "reasonable" composition:



(日) (四) (日) (日) (日)

We can represent this by a formal 2-simplex

If we give a "reasonable" composition:



(日) (四) (日) (日) (日)

We can represent this by a formal 2-simplex but it is still somewhat ill defined

# Problem with definition in general



▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 のへで

Not just the composition is undefined:

## Problem with definition in general



(日) (四) (日) (日) (日)

Not just the composition is undefined:

objects are undefined (who am I?)
# Problem with definition in general



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

Not just the composition is undefined:

- objects are undefined (who am I?)
- relations are undefined (what does "friend of" mean)

### Problem with definition in general



Not just the composition is undefined:

- objects are undefined (who am I?)
- relations are undefined (what does "friend of" mean)
- composites are undefined (how does "friend of a friend" imply "can contact")

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

As humans, we have the ability to elaborate on concepts

As humans, we have the ability to elaborate on concepts I can elaborate upon myself

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

As humans, we have the ability to elaborate on concepts I can elaborate upon myself



As humans, we have the ability to elaborate on concepts I can elaborate upon Alice



As humans, we have the ability to elaborate on concepts I can elaborate upon Alice



As humans, we have the ability to elaborate on concepts I can elaborate upon Alice





As humans, we have the ability to elaborate on concepts I can elaborate upon "Friend"

 $\mathsf{Me} \stackrel{\mathsf{friend}}{\to} \mathsf{Alice}$ 



#### As humans, we have the ability to elaborate on concepts Consider the proposed composition



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

# Elaboration of Composites



# Elaboration of Composites









◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで





- Concrete ideas are part of these elaborations.
- Elaborations are subconcepts which "define" the idea

・ロト ・ 国 ト ・ ヨ ト ・ ヨ ト

э

This is not a new idea at all:



#### This is not a new idea at all: Spaces $\rightarrow$ Open Sets

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

This is not a new idea at all: Spaces  $\rightarrow$  Open Sets We can "expand" a space to its category of open sets.

 $O:\mathfrak{Top}^{op}\to\mathfrak{Cat}$ 

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三 のへぐ

This is not a new idea at all: Spaces  $\rightarrow$  Open Sets We can "expand" a space to its category of open sets.

 $O:\mathfrak{Top}^{op}\to\mathfrak{Cat}$ 

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

Goal 2

To define a process of elaborating upon data

# Ontological Expansion: Open Sets

Let's rework this expansion



Let's rework this expansion, for simplicity:

Consider the graph of topological spaces (bounded in cardinality)

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Let's rework this expansion, for simplicity:

Consider the graph of topological spaces (bounded in cardinality)

• For each space X, let  $O(X) = \{U \text{ open in } X; i : U \rightarrow U'\}$ 

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

### Ontological Expansion: Open Sets

Let's rework this expansion, for simplicity:

Consider the graph of topological spaces (bounded in cardinality)

- For each space X, let  $O(X) = \{U \text{ open in } X; i : U \rightarrow U'\}$
- For a map  $f: X \to Y$  let  $O(f) = \{f|_U : U \to V | f(U) \subseteq V\}$

### Ontological Expansion: Open Sets

Let's rework this expansion, for simplicity:

Consider the graph of topological spaces (bounded in cardinality)

- For each space X, let  $O(X) = \{U \text{ open in } X; i : U \rightarrow U'\}$
- For a map  $f: X \to Y$  let  $O(f) = \{f|_U : U \to V | f(U) \subseteq V\}$



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

(ロ)、(型)、(E)、(E)、 E) の(()

O(X) is a sub-graph of Top, call this a 0-submorphism

O(X) is a sub-graph of Top, call this a 0-submorphism

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

O(f) is a subset of edges, call this a 1-submorphism

O(X) is a sub-graph of Top, call this a 0-submorphism

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

• O(f) is a subset of edges, call this a 1-submorphism

We are expanding our concepts into "submorphisms"

- O(X) is a sub-graph of Top, call this a 0-submorphism
- O(f) is a subset of edges, call this a 1-submorphism

We are expanding our concepts into "submorphisms"



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

#### A graph $\Sigma$ is actually a functor $\Sigma: G^{op} \to \mathfrak{Set}$

A graph  $\Sigma$  is actually a functor  $\Sigma: G^{op} \to \mathfrak{Set}$ , a "basic ontology"

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

A graph  $\Sigma$  is actually a functor  $\Sigma : G^{op} \to \mathfrak{Set}$ , a "basic ontology" • for each object of  $G = \{0 \stackrel{s}{\underset{t}{\Longrightarrow}} 1\}$ , let's <u>choose</u> a subcategory:

A graph  $\Sigma$  is actually a functor  $\Sigma : G^{op} \to \mathfrak{Set}$ , a "basic ontology" • for each object of  $G = \{0 \stackrel{s}{\underset{t}{\Longrightarrow}} 1\}$ , let's <u>choose</u> a subcategory:  $G_0 = \{0 \stackrel{s}{\underset{t}{\Longrightarrow}} 1\}$  and

A graph  $\Sigma$  is actually a functor  $\Sigma : G^{op} \to \mathfrak{Set}$ , a "basic ontology" • for each object of  $G = \{0 \stackrel{s}{\underset{t}{\Longrightarrow}} 1\}$ , let's <u>choose</u> a subcategory:  $G_0 = \{0 \stackrel{s}{\underset{t}{\Longrightarrow}} 1\}$  and  $G_1 = \{1\}$ 

A graph  $\Sigma$  is actually a functor  $\Sigma : G^{op} \to \mathfrak{Set}$ , a "basic ontology" • for each object of  $G = \{0 \stackrel{s}{\underset{t}{\Longrightarrow}} 1\}$ , let's <u>choose</u> a subcategory:  $G_0 = \{0 \stackrel{s}{\underset{t}{\Longrightarrow}} 1\}$  and  $G_1 = \{1\}$ 

• These come with inclusions  $J_0: G_0 \rightarrow G$  and  $J_1: G_1 \rightarrow G$
A graph  $\Sigma$  is actually a functor  $\Sigma : G^{op} \to \mathfrak{Set}$ , a "basic ontology" • for each object of  $G = \{0 \stackrel{s}{\underset{t}{\Rightarrow}} 1\}$ , let's <u>choose</u> a subcategory:  $G_0 = \{0 \stackrel{s}{\underset{t}{\Rightarrow}} 1\}$  and  $G_1 = \{1\}$ 

• These come with inclusions  $J_0: G_0 \to G$  and  $J_1: G_1 \to G$ We can restrict our graph to these subcategories:

A graph  $\Sigma$  is actually a functor  $\Sigma : G^{op} \to \mathfrak{Set}$ , a "basic ontology" • for each object of  $G = \{0 \stackrel{s}{\underset{t}{\Longrightarrow}} 1\}$ , let's <u>choose</u> a subcategory:  $G_0 = \{0 \stackrel{s}{\underset{t}{\Rightarrow}} 1\}$  and  $G_1 = \{1\}$ 

• These come with inclusions  $J_0 : G_0 \to G$  and  $J_1 : G_1 \to G$ We can restrict our graph to these subcategories:



A graph  $\Sigma$  is actually a functor  $\Sigma : G^{op} \to \mathfrak{Set}$ , a "basic ontology" • for each object of  $G = \{0 \stackrel{s}{\underset{t}{\Longrightarrow}} 1\}$ , let's <u>choose</u> a subcategory:  $G_0 = \{0 \stackrel{s}{\underset{t}{\Rightarrow}} 1\}$  and  $G_1 = \{1\}$ 

• These come with inclusions  $J_0 : G_0 \to G$  and  $J_1 : G_1 \to G$ We can restrict our graph to these subcategories:



### Graph Submorphism

an n-submorphism of a graph  $\Sigma : G^{op} \to \mathfrak{Set}$  is a pair  $S_n = (I_n, \eta)$ .

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

• 
$$I_n: G_n^{op} \to \mathfrak{Set}$$
, a functor

• 
$$\eta: I_n \to \Sigma \circ J_n^{op}$$
, a natural transformation

#### Graph Submorphism

an n-submorphism of a graph  $\Sigma : G^{op} \to \mathfrak{Set}$  is a pair  $S_n = (I_n, \eta)$ .

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ ▲ 三 ● ● ●

• 
$$I_n: G_n^{op} \to \mathfrak{Set}$$
, a functor  
•  $\eta: I_n \to \Sigma \circ J_n^{op}$ , a natural transformation



#### Graph Submorphism

an n-submorphism of a graph  $\Sigma : G^{op} \to \mathfrak{Set}$  is a pair  $S_n = (I_n, \eta)$ .

• 
$$I_n: G_n^{op} \to \mathfrak{Set}$$
, a functor  
•  $\eta: I_n \to \Sigma \circ J_n^{op}$ , a natural transformation



0-submorphisms are the subgraphs

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

#### Graph Submorphism

an n-submorphism of a graph  $\Sigma : G^{op} \to \mathfrak{Set}$  is a pair  $S_n = (I_n, \eta)$ .

• 
$$I_n: G_n^{op} \to \mathfrak{Set}$$
, a functor  
•  $\eta: I_n \to \Sigma \circ J_n^{op}$ , a natural transformation



- 0-submorphisms are the subgraphs
- 1-submorphisms are subsets of edges

### Graph Submorphism

an n-submorphism of a graph  $\Sigma : G^{op} \to \mathfrak{Set}$  is a pair  $S_n = (I_n, \eta)$ .

• 
$$I_n: G_n^{op} \to \mathfrak{Set}$$
, a functor  
•  $\eta: I_n \to \Sigma \circ J_n^{op}$ , a natural transformation



- 0-submorphisms are the subgraphs
- 1-submorphisms are subsets of edges

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

n-submorphisms form a category  $sm(\Sigma)[n]$ 

## For a general basic ontology $\Sigma:\Delta^{op}\to \mathfrak{C}$ we do the same

For a general basic ontology  $\Sigma : \Delta^{op} \to \mathfrak{C}$  we do the same The "choice of subcategories"  $\Delta_n \hookrightarrow \Delta$  is a functor

$$[]: \Delta \rightarrow \mathit{Fun}(*, \Delta)[2]$$

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

For a general basic ontology  $\Sigma : \Delta^{op} \to \mathfrak{C}$  we do the same The "choice of subcategories"  $\Delta_n \hookrightarrow \Delta$  is a functor

$$[]: \Delta \to \mathit{Fun}(*, \Delta)[2]$$



イロト 不得 トイヨト イヨト

3

# General Submorphisms

### n-Submorphism

an n-submorphism of a basic ontology  $\Sigma : \Delta^{op} \to \mathfrak{C}$  is a pair  $S_n = (I_n, \eta).$   $I_n : \Delta_n^{op} \to \mathfrak{C}$  a functor  $\eta : I_n \to \Sigma \circ J_n^{op}$  a natural transformation



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

# General Submorphisms

### n-Submorphism

an n-submorphism of a basic ontology  $\Sigma : \Delta^{op} \to \mathfrak{C}$  is a pair  $S_n = (I_n, \eta).$   $I_n : \Delta_n^{op} \to \mathfrak{C}$  a functor  $\eta : I_n \to \Sigma \circ J_n^{op}$  a natural transformation



▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

 $sm(\Sigma)[n]$  forms an entire category

# General Submorphisms

#### n-Submorphism

an n-submorphism of a basic ontology  $\Sigma : \Delta^{op} \to \mathfrak{C}$  is a pair  $S_n = (I_n, \eta).$   $I_n : \Delta_n^{op} \to \mathfrak{C}$  a functor  $\eta : I_n \to \Sigma \circ J_n^{op}$  a natural transformation



 $sm(\Sigma)[n]$  forms an entire category  $(Fun(\Delta_n^{op}, \mathfrak{C})/\Sigma \circ J_n^{op})$ 



We draw 1-submorphisms as if there is a domain and codomain

◆□▶ ◆□▶ ◆ 臣▶ ◆ 臣▶ ○ 臣 ○ の Q @



We draw 1-submorphisms as if there is a domain and codomain But this is not yet rigorous.

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ



We draw 1-submorphisms as if there is a domain and codomain But this is not yet rigorous. Informally:  $S_0$  is <u>a</u> domain of  $S_1$  if

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三 のへぐ



We draw 1-submorphisms as if there is a domain and codomain But this is not yet rigorous. Informally:  $S_0$  is <u>a</u> domain of  $S_1$  if

$$s \in S_1 \implies dom(s) \in S_0$$

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三 のへぐ



We draw 1-submorphisms as if there is a domain and codomain But this is not yet rigorous. Informally:  $S_0$  is <u>a</u> domain of  $S_1$  if

$$s \in S_1 \implies \textit{dom}(s) \in S_0$$

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

The 1-submorphism  $S_1$  has an entire subcategory of 0-submorphisms domains



We draw 1-submorphisms as if there is a domain and codomain But this is not yet rigorous. Informally:  $S_0$  is <u>a</u> domain of  $S_1$  if

$$s \in S_1 \implies \textit{dom}(s) \in S_0$$

The 1-submorphism  $S_1$  has an entire subcategory of 0-submorphisms domains

$$\widehat{dom}(S_1) \hookrightarrow sm(\Sigma)[0]$$

・ロト ・ 国 ト ・ ヨ ト ・ ヨ ト

э

## **Formal Faces**

Inuitively, this diagram is saying:

"Restrict the elements of  $S_n$  to the n-elements  $\sigma^n \in S_n \cap \Delta[n]$  and check that  $f(\sigma^n) \in S_k$ "

▲□▶▲□▶▲≡▶▲≡▶ ≡ めぬぐ



We reformulate our criteria as a pullback:





We reformulate our criteria as a pullback:

$$\begin{split} \hat{f}(S_n) & \longleftrightarrow sm(\Sigma)[k] \\ \downarrow & \downarrow^{k^*} \\ \Sigma_*(f) \circ_1 n^*(S_n)/\mathbb{Q}_k & \longleftrightarrow \mathbb{Q}_k \\ (\text{Where } \mathbb{Q}_k = \mathfrak{Cat}(*,\mathfrak{C})/k^*(J_k^*(\Sigma)) \end{split}$$

We reformulate our criteria as a pullback:

forgetting about the details: we get a subcategory

$$\widehat{f}(S_n) \hookrightarrow sm(\Sigma)[k]$$

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

This pullback allows us to prove basic properties of sm

$$\hat{f}: sm(\Sigma)[n] \rightarrow P(sm(\Sigma)[k])$$

## $\hat{f} : sm(\Sigma)[n] \rightarrow P(sm(\Sigma)[k])$ (where $P : \mathfrak{Cat} \rightarrow \mathfrak{Cat}$ is the subcategory monad)

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

# $\hat{f}: sm(\Sigma)[n] \rightarrow P(sm(\Sigma)[k])$

(where  $P : \mathfrak{Cat} \to \mathfrak{Cat}$  is the subcategory monad)

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

•  $sm(\Sigma)[n]$  are the n-submorphisms

# $\hat{f}: sm(\Sigma)[n] \rightarrow P(sm(\Sigma)[k])$

(where  $P : \mathfrak{Cat} \to \mathfrak{Cat}$  is the subcategory monad)

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

- $sm(\Sigma)[n]$  are the n-submorphisms
- $sm(\Sigma)[f]$  gives the face sub-categories

# $\hat{f}: sm(\Sigma)[n] \rightarrow P(sm(\Sigma)[k])$

(where  $P : \mathfrak{Cat} \to \mathfrak{Cat}$  is the subcategory monad)

- $sm(\Sigma)[n]$  are the n-submorphisms
- sm(Σ)[f] gives the face sub-categories

### Proposition

 $sm(\Sigma): \Delta^{op} \to \mathfrak{Cat}_P$  is a Lax Functor

(Where  $\mathfrak{Cat}_P$  is the Kleisli Category)

#### Moreover:

#### Theorem

 $\mathit{sm}: \mathit{Fun}(\Delta^{op}, \mathfrak{C}) 
ightarrow \mathit{Lax}(\Delta^{op}, \mathfrak{Cat}_P)$  is a functor

That is, from any basic ontology  $\Sigma$ , we get the "expansions", the submorphisms  $sm(\Sigma)$ 



• concepts are organized in a Basic Ontology  $\Sigma: \Delta^{op} o \mathfrak{C}$ 



• concepts are organized in a Basic Ontology  $\Sigma : \Delta^{op} \to \mathfrak{C}$ 

 expansions of concepts are submorphisms sm(Σ) : Δ<sup>op</sup> → Cat<sub>P</sub>



• concepts are organized in a Basic Ontology  $\Sigma : \Delta^{op} \to \mathfrak{C}$ 

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

- expansions of concepts are submorphisms  $sm(\Sigma): \Delta^{op} \to \mathfrak{Cat}_P$
- We want to expand concepts into submorphisms



 $\blacksquare$  concepts are organized in a Basic Ontology  $\Sigma:\Delta^{op}\to \mathfrak{C}$ 

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

• expansions of concepts are submorphisms  $sm(\Sigma): \Delta^{op} \to \mathfrak{Cat}_P$ 

• We want to expand concepts into submorphisms Niavely an Ontological Expansion would be a natural

 $O:\Sigma\to \textit{sm}(\mathcal{T})$ 



- concepts are organized in a Basic Ontology  $\Sigma: \Delta^{op} \to \mathfrak{C}$
- expansions of concepts are submorphisms  $sm(\Sigma): \Delta^{op} \to \mathfrak{Cat}_P$

We want to expand concepts into submorphisms
 Niavely an Ontological Expansion would be a natural



▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

# Rigorous Ontological expansion in $\mathfrak{C}=\mathfrak{Set}$

However,  $\Sigma : \Delta^{op} \to \mathfrak{Set}$ 


However,  $\Sigma : \Delta^{op} \to \mathfrak{Set}$ , but  $sm(\mathcal{T}) : \Delta^{op} \to \mathfrak{Cat}_P$ 

However,  $\Sigma : \Delta^{op} \to \mathfrak{Set}$ , but  $sm(\mathcal{T}) : \Delta^{op} \to \mathfrak{Cat}_P$ 

 For 𝔅 = 𝔅𝔅𝔅, we lift Σ : Δ<sup>op</sup> → 𝔅𝔅𝔅 "trivially" to a functor tr(Σ) : Δ<sup>op</sup> → 𝔅𝔅𝔅<sub>P</sub>

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

However,  $\Sigma : \Delta^{op} \to \mathfrak{Set}$ , but  $sm(\mathcal{T}) : \Delta^{op} \to \mathfrak{Cat}_P$ 

• For  $\mathfrak{C} = \mathfrak{Set}$ , we lift  $\Sigma : \Delta^{op} \to \mathfrak{Set}$  "trivially" to a functor  $tr(\Sigma) : \Delta^{op} \to \mathfrak{Cat}_P$ 

▲□▶ ▲□▶ ▲ □▶ ▲ □▶ □ のへぐ

■ in Get we compose with the free "discreet" functor

However,  $\Sigma : \Delta^{op} \to \mathfrak{Set}$ , but  $sm(\mathcal{T}) : \Delta^{op} \to \mathfrak{Cat}_P$ 

- For  $\mathfrak{C} = \mathfrak{Set}$ , we lift  $\Sigma : \Delta^{op} \to \mathfrak{Set}$  "trivially" to a functor  $tr(\Sigma) : \Delta^{op} \to \mathfrak{Cat}_P$
- in Get we compose with the free "discreet" functor

 $Fr:\mathfrak{Set}\to\mathfrak{Cat}$ 

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

However,  $\Sigma : \Delta^{op} \to \mathfrak{Set}$ , but  $sm(\mathcal{T}) : \Delta^{op} \to \mathfrak{Cat}_P$ 

- For  $\mathfrak{C} = \mathfrak{Set}$ , we lift  $\Sigma : \Delta^{op} \to \mathfrak{Set}$  "trivially" to a functor  $tr(\Sigma) : \Delta^{op} \to \mathfrak{Cat}_P$
- in Get we compose with the free "discreet" functor

 $Fr:\mathfrak{Set}\to\mathfrak{Cat}$ 

and the kleisli inclusion  $\{\} : \mathfrak{Cat} \to \mathfrak{Cat}_P$ 

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

However,  $\Sigma : \Delta^{op} \to \mathfrak{Set}$ , but  $sm(\mathcal{T}) : \Delta^{op} \to \mathfrak{Cat}_P$ 

- For  $\mathfrak{C} = \mathfrak{Set}$ , we lift  $\Sigma : \Delta^{op} \to \mathfrak{Set}$  "trivially" to a functor  $tr(\Sigma) : \Delta^{op} \to \mathfrak{Cat}_P$
- in Set we compose with the free "discreet" functor

 $Fr:\mathfrak{Set}\to\mathfrak{Cat}$ 

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ● ●

and the kleisli inclusion  $\{\} : \mathfrak{Cat} \to \mathfrak{Cat}_P$  $tr(\Sigma) = \{Fr \circ \Sigma\} : \Delta^{op} \to \mathfrak{Cat}_P$ 

However,  $\Sigma : \Delta^{op} \to \mathfrak{Set}$ , but  $sm(\mathcal{T}) : \Delta^{op} \to \mathfrak{Cat}_P$ 

- For  $\mathfrak{C} = \mathfrak{Set}$ , we lift  $\Sigma : \Delta^{op} \to \mathfrak{Set}$  "trivially" to a functor  $tr(\Sigma) : \Delta^{op} \to \mathfrak{Cat}_P$
- in 𝔅𝔄t we compose with the free "discreet" functor

 $Fr:\mathfrak{Set}\to\mathfrak{Cat}$ 

and the kleisli inclusion  $\{\} : \mathfrak{Cat} \to \mathfrak{Cat}_P$ 

- $tr(\Sigma) = {Fr \circ \Sigma} : \Delta^{op} \to \mathfrak{Cat}_P$
- this gives us a (representable) functor

 $tr: Fun(\Delta^{op}, \mathfrak{Set}) \to Lax(\Delta^{op}, \mathfrak{Cat}_P)$ 

In set, we have:



In set, we have:

• "Trivial" expansion  $tr: Fun(\Delta^{op}, \mathfrak{Set}) \to Lax(\Delta^{op}, \mathfrak{Cat}_P)$ 

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

In set, we have:

• "Trivial" expansion  $tr: Fun(\Delta^{op}, \mathfrak{Set}) \to Lax(\Delta^{op}, \mathfrak{Cat}_P)$ 

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

"Real" expansion functor
 sm : Fun(Δ<sup>op</sup>, Get) → Lax(Δ<sup>op</sup>, Cat<sub>P</sub>)

In set, we have:

- "Trivial" expansion  $tr: Fun(\Delta^{op}, \mathfrak{Set}) \to Lax(\Delta^{op}, \mathfrak{Cat}_P)$
- "Real" expansion functor
   sm : Fun(Δ<sup>op</sup>, Get) → Lax(Δ<sup>op</sup>, Cat<sub>P</sub>)
- two basic ontologies  $\Sigma, \mathcal{T} : \Delta^{op} \to \mathfrak{Set}$

#### Ontological Expansion

an Ontological Expansion is a natural transformation

$$O: tr(\Sigma) \to sm(\mathcal{T})$$

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ● ●



#### What is $tr : Fun(\Delta^{op}, \mathfrak{C}) \to Lax(\Delta^{op}, \mathfrak{Cat}_P)$ for general $\mathfrak{C}$ ?

(ロ)、(型)、(E)、(E)、(E)、(O)へ(C)

#### What is $tr : Fun(\Delta^{op}, \mathfrak{C}) \to Lax(\Delta^{op}, \mathfrak{Cat}_P)$ for general $\mathfrak{C}$ ? The answer might lie in the 3rd goal

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

#### What is $tr : Fun(\Delta^{op}, \mathfrak{C}) \to Lax(\Delta^{op}, \mathfrak{Cat}_P)$ for general $\mathfrak{C}$ ? The answer might lie in the 3rd goal

#### Goal 3

Reconstruct data from it's elaborations

we're going to take cues from Grothendieck Topologies

・ロト・西ト・山田・山田・山口・

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Category to organize concepts

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

- Category to organize concepts
- Coverings are elaborations

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

- Category to organize concepts
- Coverings are elaborations
- Sheaf condition reconstructs data



#### "a covering of a covering is a covering"



"a covering of a covering is a covering"  $\rightarrow$  "An expansion of an expansion is an expansion"

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

- "a covering of a covering is a covering"
- $\rightarrow$  "An expansion of an expansion is an expansion"

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 のへで

 $\rightarrow$  composition of expansion

- "a covering of a covering is a covering"
- $\rightarrow$  "An expansion of an expansion is an expansion"
- $\rightarrow$  composition of expansion
  - From  $O: tr(\Sigma) \to sm(\Sigma')$  and  $O': tr(\Sigma') \to sm(\Sigma'')$

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● の Q @

- "a covering of a covering is a covering"
- $\rightarrow$  "An expansion of an expansion is an expansion"
- $\rightarrow$  composition of expansion

• From  $O: tr(\Sigma) \to sm(\Sigma')$  and  $O': tr(\Sigma') \to sm(\Sigma'')$ we want an  $O' \circ O: tr(\Sigma) \to sm(\Sigma'')$ 

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQで

#### Conjecture

sm:  $(\Delta^{op}/2 - \mathfrak{Cat})_{Lax} \rightarrow (\Delta^{op}/2 - \mathfrak{Cat})_{Lax}$ 



#### Conjecture

sm: 
$$(\Delta^{op}/2 - \mathfrak{Cat})_{Lax} \rightarrow (\Delta^{op}/2 - \mathfrak{Cat})_{Lax}$$

$$tr(\Sigma') \xrightarrow{O'} sm(\Sigma'')$$

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

$$tr(\Sigma) \xrightarrow{O} sm(\Sigma')$$

#### Conjecture

sm: 
$$(\Delta^{op}/2 - \mathfrak{Cat})_{Lax} \rightarrow (\Delta^{op}/2 - \mathfrak{Cat})_{Lax}$$

$$\begin{split} tr(\Sigma') &\xrightarrow{O'} sm(\Sigma'') \\ & \clubsuit \\ tr(\Sigma) &\xrightarrow{O} sm(\Sigma') \quad sm(tr(\Sigma')) \xrightarrow{sm(O')} sm(sm(\Sigma'')) \end{split}$$

◆□ ▶ ◆□ ▶ ◆ □ ▶ ◆ □ ▶ ○ □ ○ ○ ○ ○

#### Conjecture

sm: 
$$(\Delta^{op}/2 - \mathfrak{Cat})_{Lax} \rightarrow (\Delta^{op}/2 - \mathfrak{Cat})_{Lax}$$

$$tr(\Sigma') \xrightarrow{O'} sm(\Sigma'')$$

$$\bigcup$$

$$tr(\Sigma) \xrightarrow{O} sm(\Sigma') \xrightarrow{\iota} sm(tr(\Sigma')) \xrightarrow{sm(O')} sm(sm(\Sigma''))$$

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

#### Conjecture

sm: 
$$(\Delta^{op}/2 - \mathfrak{Cat})_{Lax} \rightarrow (\Delta^{op}/2 - \mathfrak{Cat})_{Lax}$$

$$tr(\Sigma') \xrightarrow{O'} sm(\Sigma'')$$
$$\mathbf{r}$$

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三 のへぐ

 $tr(\Sigma) \xrightarrow{O} sm(\Sigma') \xrightarrow{\iota} sm(tr(\Sigma')) \xrightarrow{sm(O')} sm(sm(\Sigma'')) \xrightarrow{\mu} sm(\Sigma'')$ 

# Current Work

$$tr(\Sigma) \stackrel{O}{\rightarrow} sm(\Sigma') \stackrel{\iota}{\rightarrow} sm(tr(\Sigma')) \stackrel{sm(O')}{\rightarrow} sm(sm(\Sigma'')) \stackrel{\mu}{\rightarrow} sm(\Sigma'')$$

## Current Work

$$tr(\Sigma) \stackrel{O}{\rightarrow} sm(\Sigma') \stackrel{\iota}{\rightarrow} sm(tr(\Sigma')) \stackrel{sm(O')}{\rightarrow} sm(sm(\Sigma'')) \stackrel{\mu}{\rightarrow} sm(\Sigma'')$$

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ 三三 - のへぐ

this gives some conditions for tr

• we have a natural  $i: tr(\Sigma') \to sm(tr(\Sigma'))$ 

## Current Work

$$tr(\Sigma) \stackrel{O}{\rightarrow} sm(\Sigma') \stackrel{\iota}{\rightarrow} sm(tr(\Sigma')) \stackrel{sm(O')}{\rightarrow} sm(sm(\Sigma'')) \stackrel{\mu}{\rightarrow} sm(\Sigma'')$$

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

this gives some conditions for tr

- we have a natural  $i : tr(\Sigma') \to sm(tr(\Sigma'))$
- seems to make sm into some sort of monad

$$tr(\Sigma) \stackrel{O}{\rightarrow} sm(\Sigma') \stackrel{\iota}{\rightarrow} sm(tr(\Sigma')) \stackrel{sm(O')}{\rightarrow} sm(sm(\Sigma'')) \stackrel{\mu}{\rightarrow} sm(\Sigma'')$$

this gives some conditions for tr

- we have a natural  $i: tr(\Sigma') \to sm(tr(\Sigma'))$
- seems to make sm into some sort of monad

if this is the case, the ontological expansions are sm-algebras

$$[(\Delta^{op}/2 - \mathfrak{Cat})_{Lax}]_{sm}$$

$$[(\Delta^{op}/2 - \mathfrak{Cat})_{Lax}]_{sm}$$

◆□▶ ◆□▶ ◆ 臣▶ ◆ 臣▶ ○ 臣 ○ の Q @

• Organize data: Objects are Basic Ontologies

$$[(\Delta^{op}/2 - \mathfrak{Cat})_{Lax}]_{sm}$$

■ Organize data: Objects are Basic Ontologies Ø



## $[(\Delta^{op}/2-\mathfrak{Cat})_{Lax}]_{sm}$

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Organize data: Objects are Basic Ontologies Ø
Elaborations: Morphism are Ontological Expansions

## $[(\Delta^{op}/2-\mathfrak{Cat})_{Lax}]_{sm}$

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

Organize data: Objects are Basic Ontologies Ø
 Elaborations: Morphism are Ontological Expansions Ø
# Proposed Definition: Ontology

## $[(\Delta^{op}/2-\mathfrak{Cat})_{Lax}]_{sm}$

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00

- Organize data: Objects are Basic Ontologies Ø
- Elaborations: Morphism are Ontological Expansions Ø
- A means of recovering data from expansions

# Proposed Definition: Ontology

## $[(\Delta^{op}/2-\mathfrak{Cat})_{Lax}]_{sm}$

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00

- Organize data: Objects are Basic Ontologies Ø
- Elaborations: Morphism are Ontological Expansions Ø
- A means of recovering data from expansions  $\Box$

### $[(\Delta^{op}/2-\mathfrak{Cat})_{Lax}]_{sm}$

- Organize data: Objects are Basic Ontologies II
- Elaborations: Morphism are Ontological Expansions ∅
- A means of recovering data from expansions □

The condition that our ontological expansions compose gives us the following definiton:

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00

# $[(\Delta^{op}/2-\mathfrak{Cat})_{Lax}]_{sm}$

- Organize data: Objects are Basic Ontologies Ø
- Elaborations: Morphism are Ontological Expansions ∅
- A means of recovering data from expansions  $\Box$

The condition that our ontological expansions compose gives us the following definiton:

▲□▶ ▲□▶ ▲□▶ ▲□▶ ■ ●の00

#### **Proposed Definition**

A  $\Delta$ -**Ontology** is a subcategory  $\mathbb{O} \hookrightarrow [(\Delta^{op}/2 - \mathfrak{Cat})_{Lax}]_{sm}$ 

intuition: [OLOG] "OLOGS: A Categorical Framework for Knowledge Representation" David I. Spivak, Robert E. Kent

(higher) category theory:

- [1] "Higher Categories, Higher Operads" Leinster
- [2] "Higher Topos Theory" Lurie
- [3] "Sheaves in Geometry and Logic" Maclane, Moerdijk
- [4] "Categories for the working mathematician" Maclane

[5] "The Stack Of Microlocal Presheaves" Waschkies