What is a *Thing*?

David Jaz Myers

Johns Hopkins University

January 4, 2019

Galileo's Argument

Suppose that heavier things fell faster than lighter ones. Then, if we tied a light stone to a heavy stone, it would slow the heavy stone down because it falls slower. But the whole thing is heavier than its parts, so it should speed up. This is a contradiction, so we know that things fall at the same speed regardless of their weight.

Galileo's Argument

> *Suppose that heavier things fell faster than lighter ones. Then, if we tied a light stone to a heavy stone, it would slow the heavy stone down because it falls slower. But the whole thing is heavier than its parts, so it should speed up. This is a contradiction, so we know that things fall at the same speed regardless of their weight.*

This argument crucially relies on what *things* are in the model.

Galileo's Argument

> *Suppose that heavier things fell faster than lighter ones. Then, if we tied a light stone to a heavy stone, it would slow the heavy stone down because it falls slower. But the whole thing is heavier than its parts, so it should speed up. This is a contradiction, so we know that things fall at the same speed regardless of their weight.*

This argument crucially relies on what *things* are in the model.

What about tying the stones together makes them *part of the same thing*?

Basic Questions

- What is a *thing*?

Basic Questions

- What is a *thing*?
- How do things come to be, and cease?

Basic Questions

- What is a *thing*?
- How do things come to be, and cease?
- How can we set up a system to make or maintain the things we want, and end the things we don't?

Things in the Sciences

- What does a chimpanzee see?

Things in the Sciences

- What does a chimpanzee see?
- What does a neural network "see"?

Things in the Sciences

- What does a chimpanzee see?
- What does a neural network “see”?
- What social groups are in active in a social network?

Things in the Sciences

- What does a chimpanzee see?
- What does a neural network "see"?
- What social groups are in active in a social network?
- What events does this climate data suggest?

Things in the Sciences

- What does a chimpanzee see?
- What does a neural network "see"?
- What social groups are in active in a social network?
- What events does this climate data suggest?
- And lots more...

Things in the Sciences

- What does a chimpanzee see?
- What does a neural network "see"?
- What social groups are in active in a social network?
- What events does this climate data suggest?
- And lots more...

- Given a model of some system, what *things* are in this model?

A Kernel of Understanding

Idea: If you pull on part of a thing, the rest will come with.

A Kernel of Understanding

Idea: If you constrain part of a thing, the rest is constrained as well.

A Kernel of Understanding

So, given the **Idea:**

> *If you constrain part of a thing, the rest is constrained as well.*

A Kernel of Understanding

So, given the **Idea:**

> *If you constrain part of a thing, the rest is constrained as well.*

The question "Is this a thing?" will be answered in terms of:

- The relationship between constraints on the parts and constraints on the whole.

The Two Noodles Thought Experiment

[noodle waving]

A Simpler Question

Question: Given a part of a system, what things is it a part of?

A Simpler Question

Question: Given a part of a system, what things is it a part of?

To answer this, we need

- A notion of "system" (or "model"),

A Simpler Question

Question: Given a part of a system, what things is it a part of?

To answer this, we need

- A notion of "system" (or "model"),
- A notion of "part",

A Simpler Question

Question: Given a part of a system, what things is it a part of?

To answer this, we need

- A notion of "system" (or "model"),
- A notion of "part",
- A notion of "constraint",

A Simpler Question

Question: Given a part of a system, what things is it a part of?

To answer this, we need

- A notion of "system" (or "model"),
- A notion of "part",
- A notion of "constraint",
- An understanding of how the constraints of some part of the system constrain other parts.

Formalizing Our Question

What should our notion of system be?

Formalizing Our Question

What should our notion of system be?

When we constrain a part of a system, we constrain *what it does*.

Formalizing Our Question

What should our notion of system be?

When we constrain a part of a system, we constrain *what it does*.

So, we should model a system by its **type of behaviors**!

What is a Behavior Type?

It is a type of behaviors (that something might do).

What is a Behavior Type?

It is a type of behaviors (that something might do).

Ok, but what exactly are they?

What is a Behavior Type?

It is a type of behaviors (that something might do).

Ok, but what exactly are they?

Whatever they are, they form a category $\mathcal{B}$! (The morphisms will be functions sending behaviors to behaviors.)

What is a Behavior Type?

It is a type of behaviors (that something might do).

Ok, but what exactly are they?

Whatever they are, they form a category $\mathcal{B}$! (The morphisms will be functions sending behaviors to behaviors.)

But we want to reason about behaviors using *logic*, so we need the category $\mathcal{B}$ of behavior types to be a *topos*.

The Briefest Introduction to Toposes

A topos is a category where you can do logic.

Definition

A topos is a category that has

- *a terminal object and pullbacks,*
- *an internal hom* $(-)^X$ *(right adjoint to* $X \times -$*).*
- *a subobject classifier* **Prop**.

The Briefest Introduction to Toposes

A topos is a category where you can do logic.

Definition

A topos is a category that has

- *a terminal object and pullbacks,*
- *an internal hom* $(-)^X$ *(right adjoint to* $X \times -$*).*
- *a subobject classifier* $\mathbf{Prop}$*.*

Given $f : X \to Y$, we get an adjoint triple:

$$\begin{tikzcd} \mathbf{Prop}^X \arrow[r, bend left, "\exists_f"] \arrow[r, bend right, "\forall_f"'] & \mathbf{Prop}^Y \arrow[l, "\Delta_f" description] \end{tikzcd}$$

What is a Part?

- If B_S is the type of possible behaviors of our system S, and P is a part of S,
- then for every behavior $s : B_S$ of S, we can see what P is doing during s, giving us a behavior $s|_P : B_P$,

What is a Part?

- If B_S is the type of possible behaviors of our system S, and P is a part of S,
- then for every behavior $s : B_S$ of S, we can see what P is doing during s, giving us a behavior $s|_P : B_P$,
- and every behavior $p : B_P$ arises in this way (since P is considered as part of S, not on its own).

What is a Part?

- If B_S is the type of possible behaviors of our system S, and P is a part of S,
- then for every behavior $s : B_S$ of S, we can see what P is doing during s, giving us a behavior $s|_P : B_P$,
- and every behavior $p : B_P$ arises in this way (since P is considered as part of S, not on its own).

Definition

If B_S is the behavior type of some system S, a part P of S is an epimorphism $|_P : B_S \twoheadrightarrow B_P$.
A part P contains Q (written $P \geq Q$) if there is an epi $|_Q : B_P \twoheadrightarrow B_Q$ so that

$$\begin{tikzcd}
& B_S \arrow[dl, two heads, "|_P"'] \arrow[dr, two heads, "|_Q"] & \\
B_P \arrow[rr, two heads, "|_Q"'] & & B_Q
\end{tikzcd}$$

Compatibility and the Lattice of Parts

Definition

Behaviors $p : B_P$ *and* $q : B_Q$ *of parts* P *and* Q *are compatible if there is a behavior* s *of the whole system which restricts to both of them:*

$$\mathfrak{c}(p, q) :\equiv \exists s : B_S .\, p = s|_P \wedge s|_Q = q.$$

Compatibility and the Lattice of Parts

Definition

Behaviors $p : B_P$ and $q : B_Q$ of parts P and Q are compatible if there is a behavior s of the whole system which restricts to both of them:

$$\mathfrak{c}(p, q) :\equiv \exists s : B_S.\, p = s|_P \wedge s|_Q = q.$$

- The *union* $B_{P \cup Q}$ of parts P and Q has behaviors given by compatible pairs of behaviors from P and from Q:

$$B_{P \cup Q} :\equiv \{(p, q) : B_P \times B_Q \mid \mathfrak{c}(p, q)\}.$$

- The *intersection* $B_{P \cap Q}$ of parts P and Q has behaviors which are either behaviors from P or from Q, but considered equal if they are compatible:

$$B_{P \cap Q} :\equiv \frac{B_P + B_Q}{\mathfrak{c}}.$$

Parts as Equivalence Relations

Given a part $B_S \twoheadrightarrow B_P$, we can consider the equivalence relation on behaviors of S

$$s \sim_P s' \Longleftrightarrow s|_Q = s'|_Q$$

that is, $s \sim_P s'$ if they involve the same behavior of Q, if "Q sees them to be the same".

Constraints

We will equate a *constraint* ϕ on the behaviors of a part P with predicate "satisfies ϕ" on B_P. That is, $\phi : B_P \to \mathbf{Prop}$.

Constraints

We will equate a *constraint* ϕ on the behaviors of a part P with predicate "satisfies ϕ" on B_P. That is, $\phi : B_P \to \mathbf{Prop}$.
Since we are in a topos, we get maps

$$\begin{tikzcd} \mathbf{Prop}^{B_S} \arrow[rr, bend left, "\exists_P"] \arrow[rr, bend right, "\forall_P"'] & & \mathbf{Prop}^{B_P} \arrow[ll, "\Delta_P"'] \end{tikzcd}$$

Constraints

We will equate a *constraint* ϕ on the behaviors of a part P with predicate "satisfies ϕ" on B_P. That is, $\phi : B_P \to \mathbf{Prop}$.
Since we are in a topos, we get maps

$$\begin{tikzcd} \mathbf{Prop}^{B_S} \arrow[rr, bend left, "\exists_P"] \arrow[rr, bend right, "\forall_P"'] & & \mathbf{Prop}^{B_P} \arrow[ll, "\Delta_P"'] \end{tikzcd}$$

A quick calculation gives:

$$\Delta_P \circ \exists_P \phi(s) = \exists s'.\, s \sim_P s' \wedge \phi(s')$$

$$\Delta_P \circ \forall_P \phi(s) = \forall s'.\, s \sim_P s' \Rightarrow \phi(s')$$

Induced Constraints

Definition

A constraint ϕ on a part P induces two interesting constraints on a part Q.

- *"Is compatible with ϕ": $\Diamond^P_Q :\equiv \exists_Q \circ \Delta_P$*

$$\Diamond^P_Q \phi(q) :\equiv \exists s : B_S.\, s|_Q = q \wedge \phi(s|_P).$$

- *"Ensures ϕ": $\Box^P_Q :\equiv \forall_Q \circ \Delta_P$*

$$\Box^P_Q \phi(q) :\equiv \forall s : B_S.\, s|_Q = q \Rightarrow \phi(s|_P).$$

Properties of Induced Constraints

Claim

- *If* $\phi \Rightarrow \psi$, *then* $\Diamond^P_Q \phi \Rightarrow \Diamond^P_Q \psi$ *and* $\Box^P_Q \phi \Rightarrow \Box^P_Q \psi$
- $\Diamond^P_P = \Box^P_P = \mathrm{id}$
- $\Box^P_Q \dashv \Diamond^Q_P$
- $\Diamond^P_Q \Rightarrow \Box^P_Q$
- $\Diamond^P_R \Rightarrow \Diamond^Q_R \circ \Diamond^P_Q$
- $\Box^Q_R \circ \Box^P_Q \Rightarrow \Box^P_R$

Properties of Induced Constraints

Claim

- $\Diamond_Q^P(\exists x.\, \phi_x) = \exists x.\, \Diamond_Q^P(\phi_x)$.
- $\Diamond_Q^P(\phi \wedge \psi) \Rightarrow \Diamond_Q^P \phi \wedge \Diamond_Q^P \psi$.
- $\Box_Q^P(\forall x.\, \phi_x) = \forall x.\, \Box_Q^P(\phi_x)$.
- $\Box_Q^P(\phi) \vee \Box_Q^P(\psi) \Rightarrow \Box_Q^P(\phi \vee \psi)$

Properties of Induced Constraints

Claim

- $\Diamond^P_{Q \cap R} \phi = \exists q : Q,\ r : R.\, \mathfrak{c}(q, r) \wedge \Diamond^P_{Q \cup R} \phi(q, r)$.
- $\Diamond^P_{Q \cup R} \phi(q, r) \Rightarrow \Diamond^P_Q \phi(q) \wedge \Diamond^P_R \phi(r)$.
- $\Box^P_{Q \cap R} \phi = \forall q : Q,\ r : R.\, \mathfrak{c}(q, r) \Rightarrow \Box^P_{Q \cup R} \phi(q, r)$.
- $\Box^P_Q \phi(q) \vee \Box^P_R \phi(r) \Rightarrow \Box^P_{Q \cup R} \phi(q, r)$.

Measuring with Numbers

Suppose we have a notion of size $\#B_P : \mathbb{R}$ for each behavior type we are considering (and their subtypes)

We can then define the *constraint ratio* for $\phi : B_P \to \mathbf{Prop}$

$$\mathrm{constr}(\phi, P) :\equiv \frac{\#B_P - \#\{\phi\}}{\#B_P}$$

as a measure of "how constrained P is by ϕ".

Then the *constraint rate* for $\phi : B_P \to \mathbf{Prop}$ and part Q

$$\mathrm{R}(\phi, Q) :\equiv \frac{\mathrm{constr}(\Diamond^P_Q \phi, Q)}{\mathrm{constr}(\phi, P)}$$

as a measure of "how constrained Q is by ϕ, relative to how constraining ϕ is".

Examples

[graph time]