Applied Compositional Thinking for Engineers

Session 13

Summary

Agenda

- Summary
- What's next to learn
- Next steps
- Feedback form

Summary: Categories and specialization

- Categories to describe **resources, transmutations**, and their **composition**

$$\langle 50, \mathsf{USD}\rangle \longrightarrow \boxed{\textbf{ExchATM}} \longrightarrow \langle ?, \mathsf{CHF}\rangle \longrightarrow \boxed{\textbf{Frankurrencies}} \longrightarrow \langle ??, \mathsf{EUR}\rangle$$

$$\langle 50, \mathsf{USD}\rangle \longrightarrow \boxed{\textbf{ExchATM}\ \mathsf{then}\ \textbf{Frankurrencies}} \longrightarrow \langle ??, \mathsf{EUR}\rangle$$

Summary: Categories and specialization

- Categories to describe **resources, transmutations,** and their **composition**

$$\langle 50, \mathsf{USD}\rangle \longrightarrow \boxed{\textbf{ExchATM}} \longrightarrow \langle ?, \mathsf{CHF}\rangle \longrightarrow \boxed{\textbf{Frankurrencies}} \longrightarrow \langle ??, \mathsf{EUR}\rangle$$

$$\langle 50, \mathsf{USD}\rangle \longrightarrow \boxed{\textbf{ExchATM}\ \mathsf{then}\ \textbf{Frankurrencies}} \longrightarrow \langle ??, \mathsf{EUR}\rangle$$

- Specialization: **subcategories**

BergAma

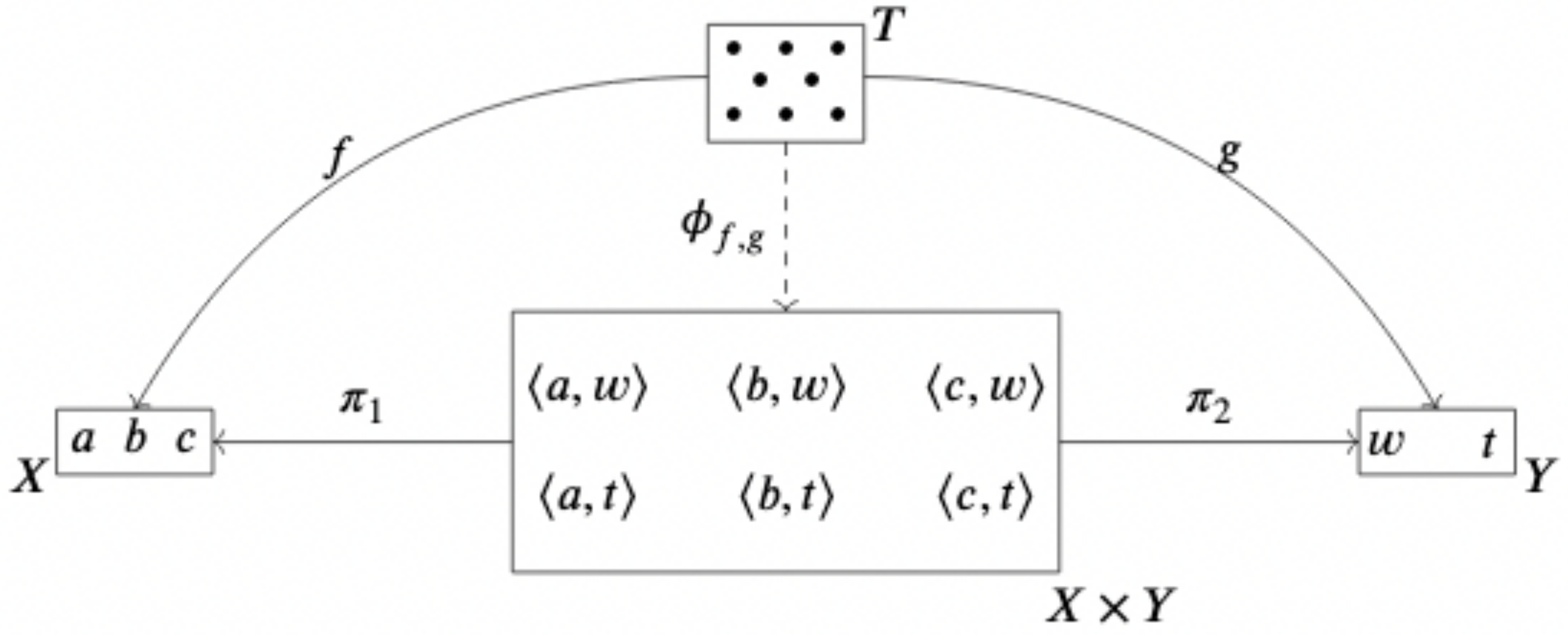
Berg

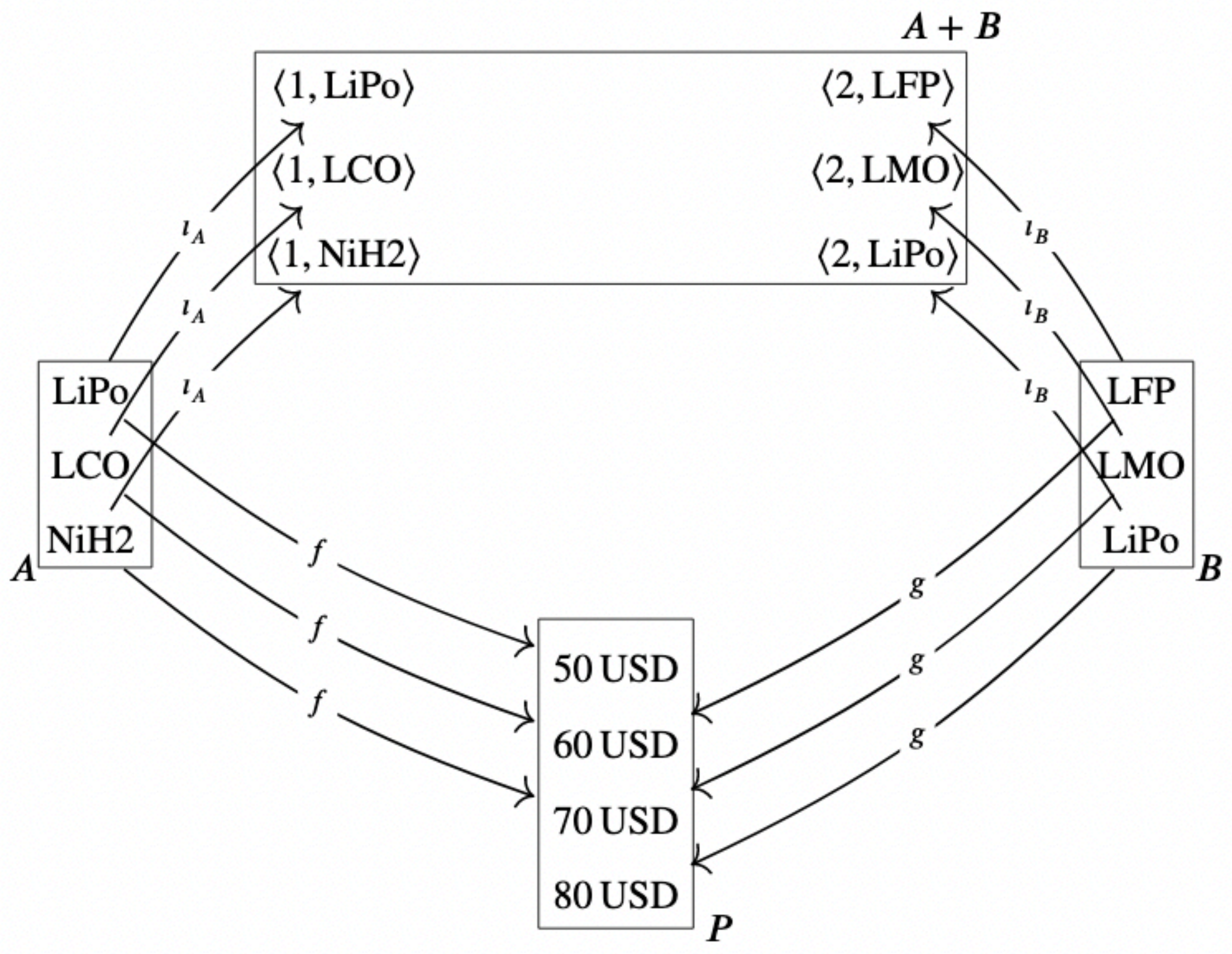
Summary: Isomorphisms, (co)products

- **Isomorphisms**

$$\begin{tikzcd} \{0,1\} \arrow[r, bend left, "f"] & \{\text{true},\ \text{false}\} \arrow[l, bend left, "g"] \end{tikzcd}$$

- **Products, coproducts**, and **universal properties**



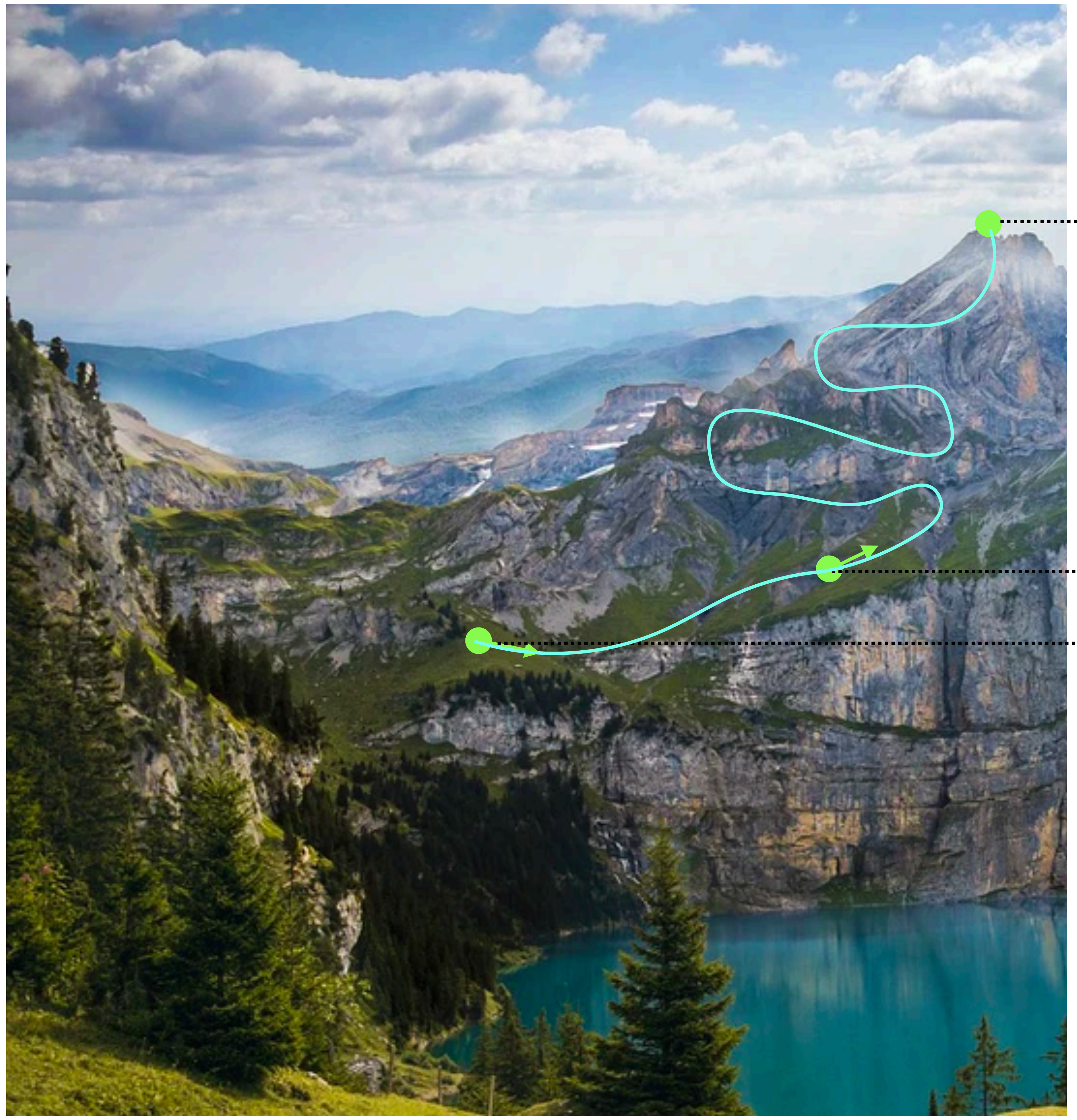


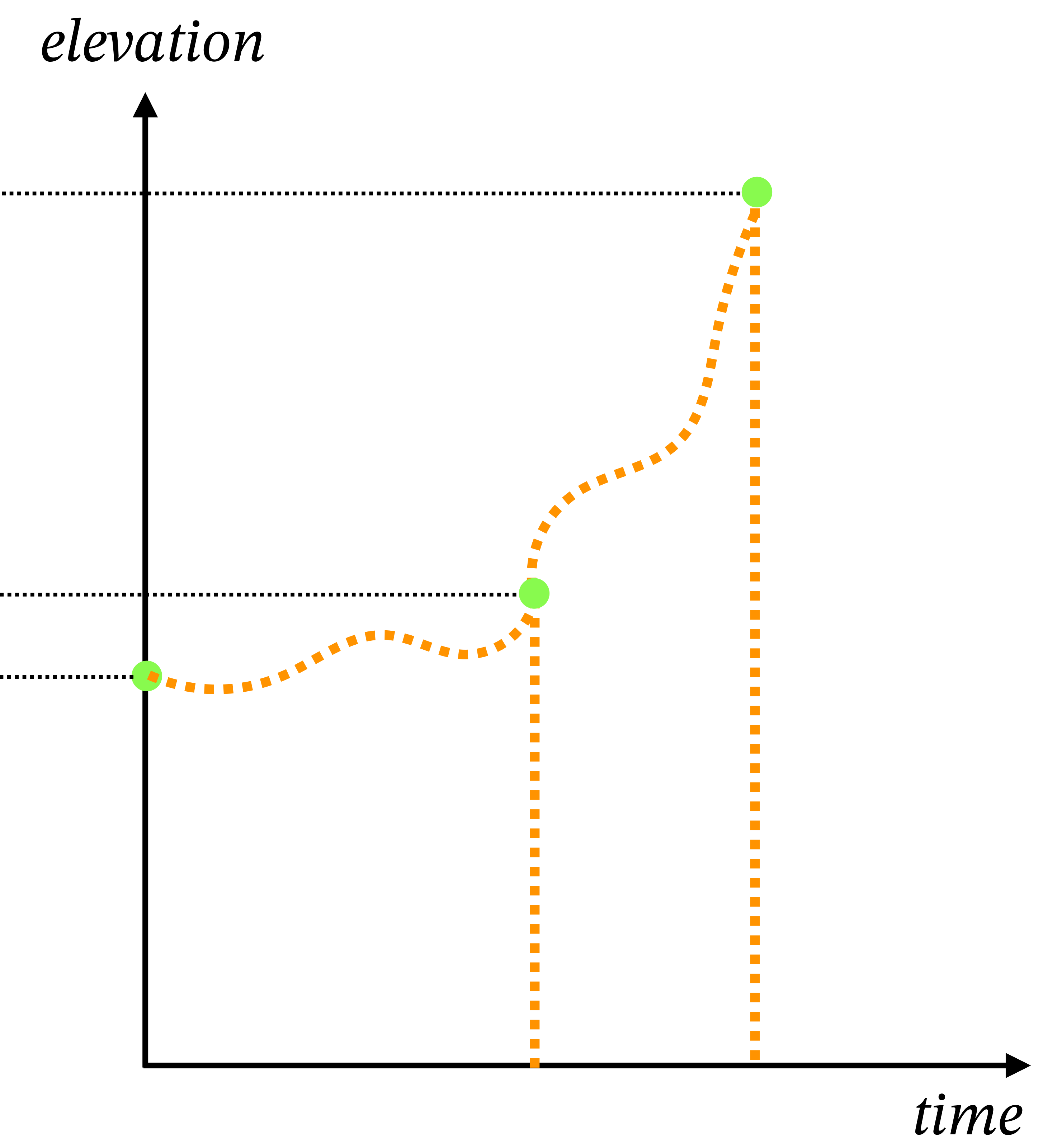
Summary: Trade-offs are everywhere

- Multiple **functionalities** and **costs**

Summary: Life is hard

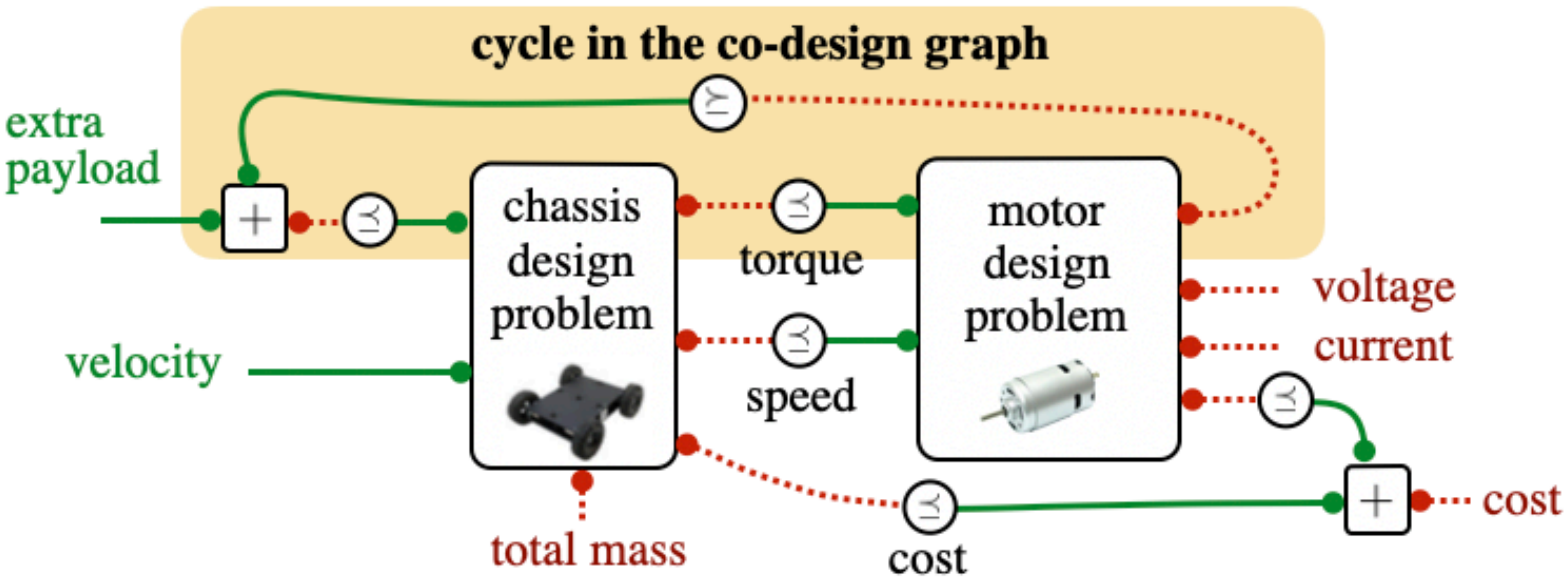
- **Functors**





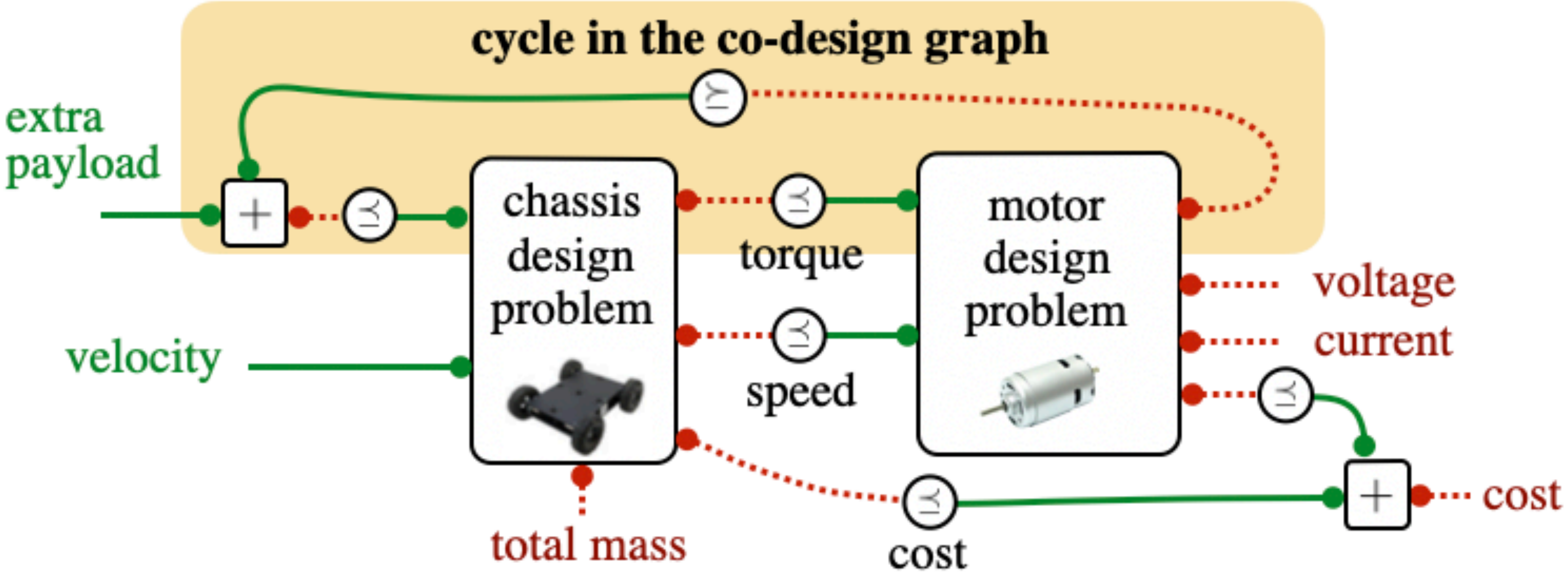
Summary: Design problems

- **Design problems** to describe:
  - (**Symmetric**) **monoidal** categories;
  - **Locally posetal** categories;
  - **Traced**, **compact closed** categories

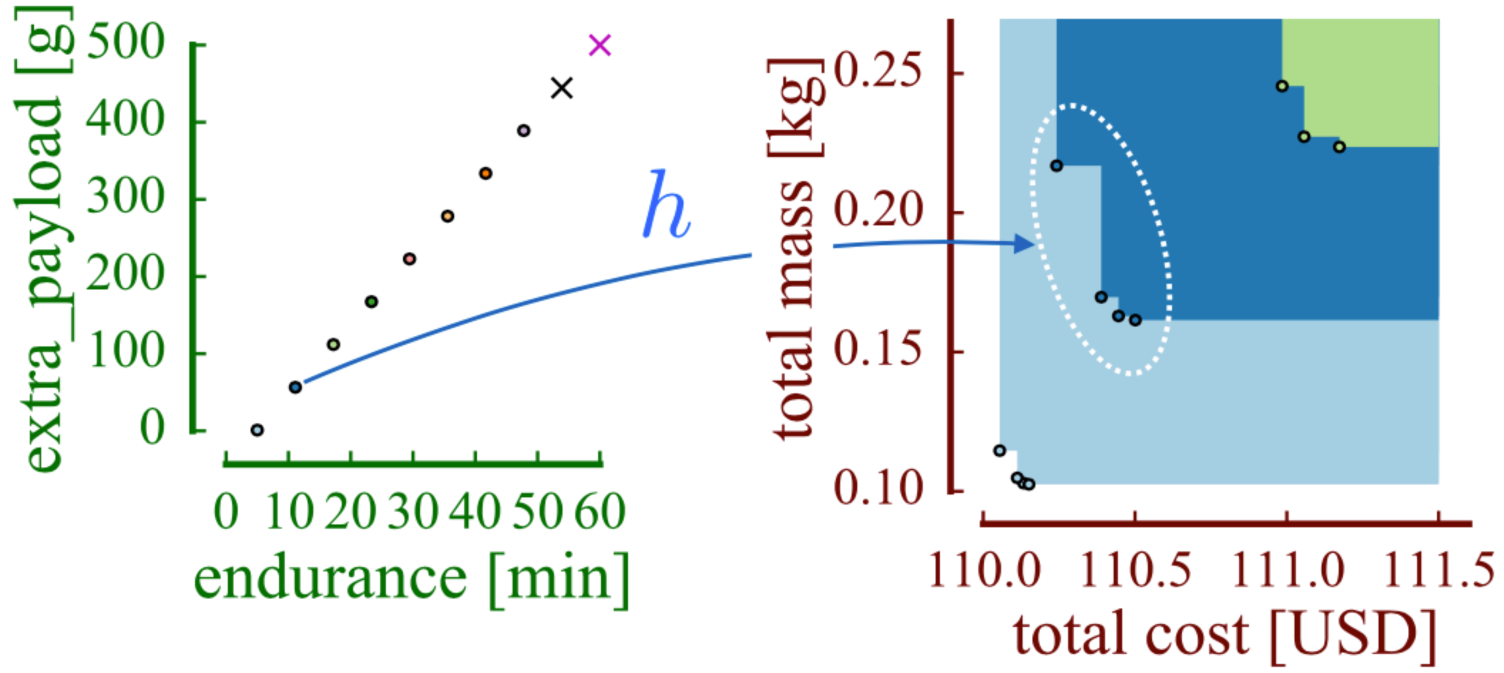


Summary: Design problems

- **Design problems** to describe:
  - (**Symmetric**) **monoidal** categories;
  - **Locally posetal** categories;
  - **Traced**, **compact closed** categories

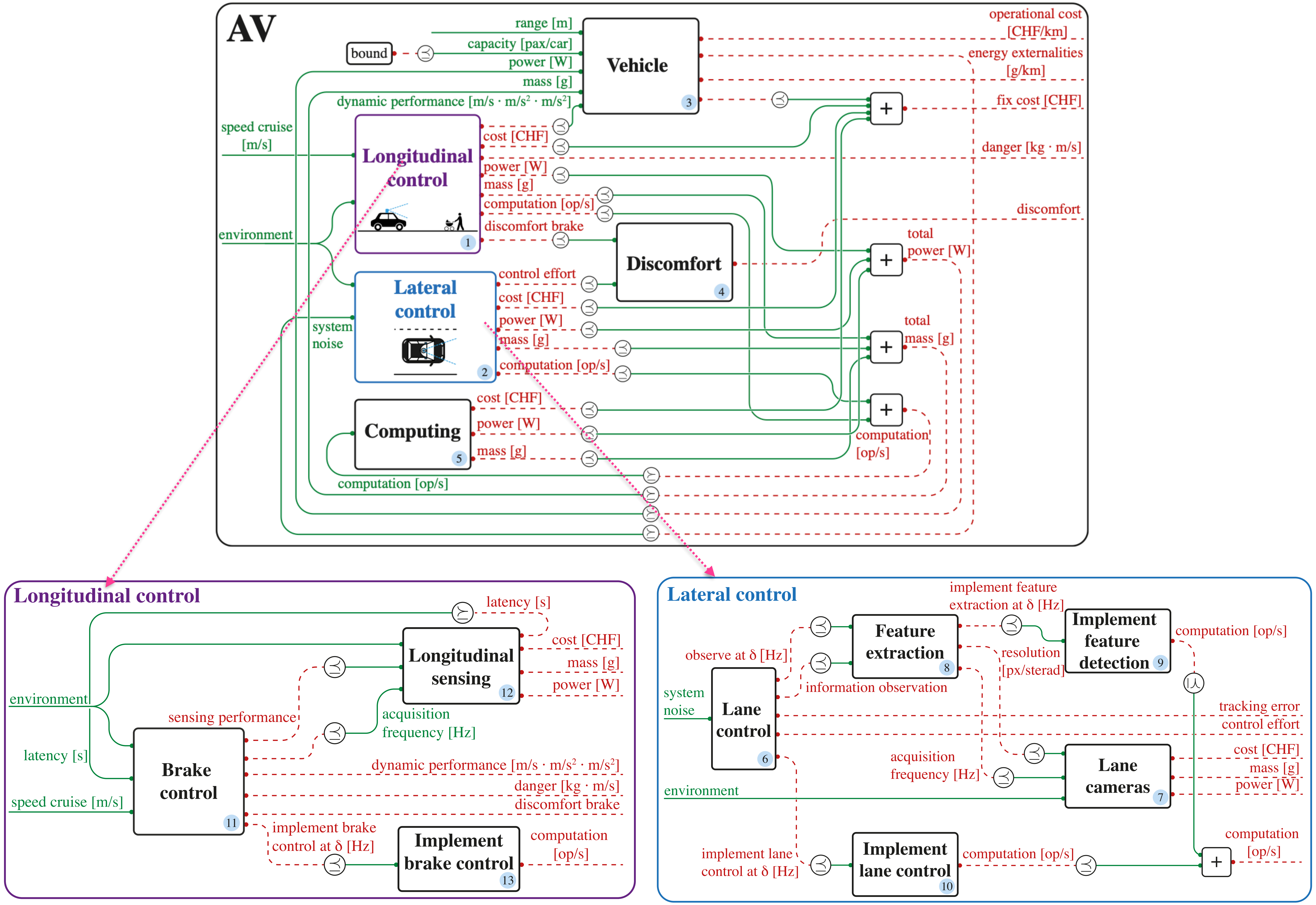


- **Computation**



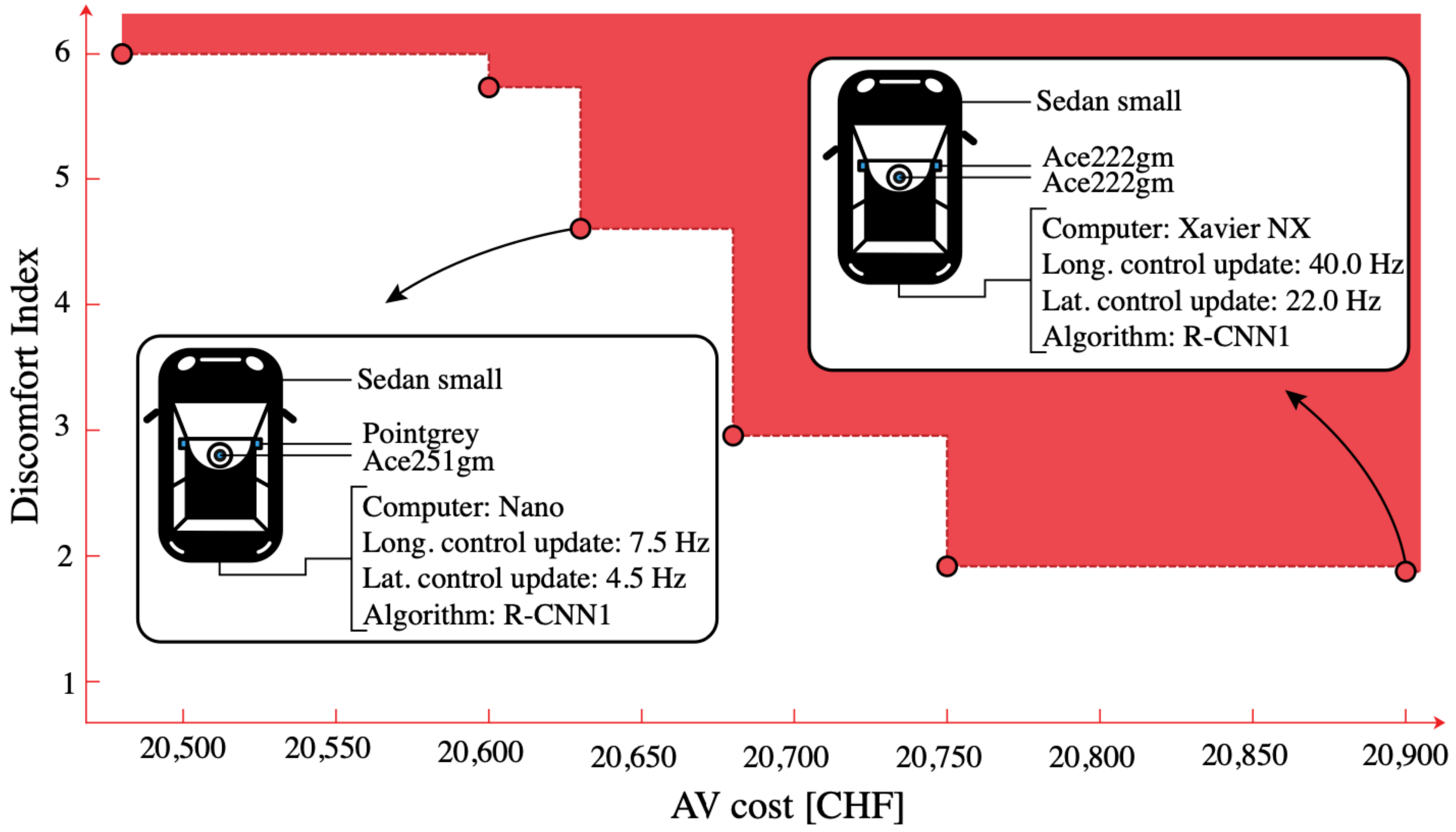
Example: Co-design of embodied intelligence

- **Co-design** of an **Autonomous Vehicle**



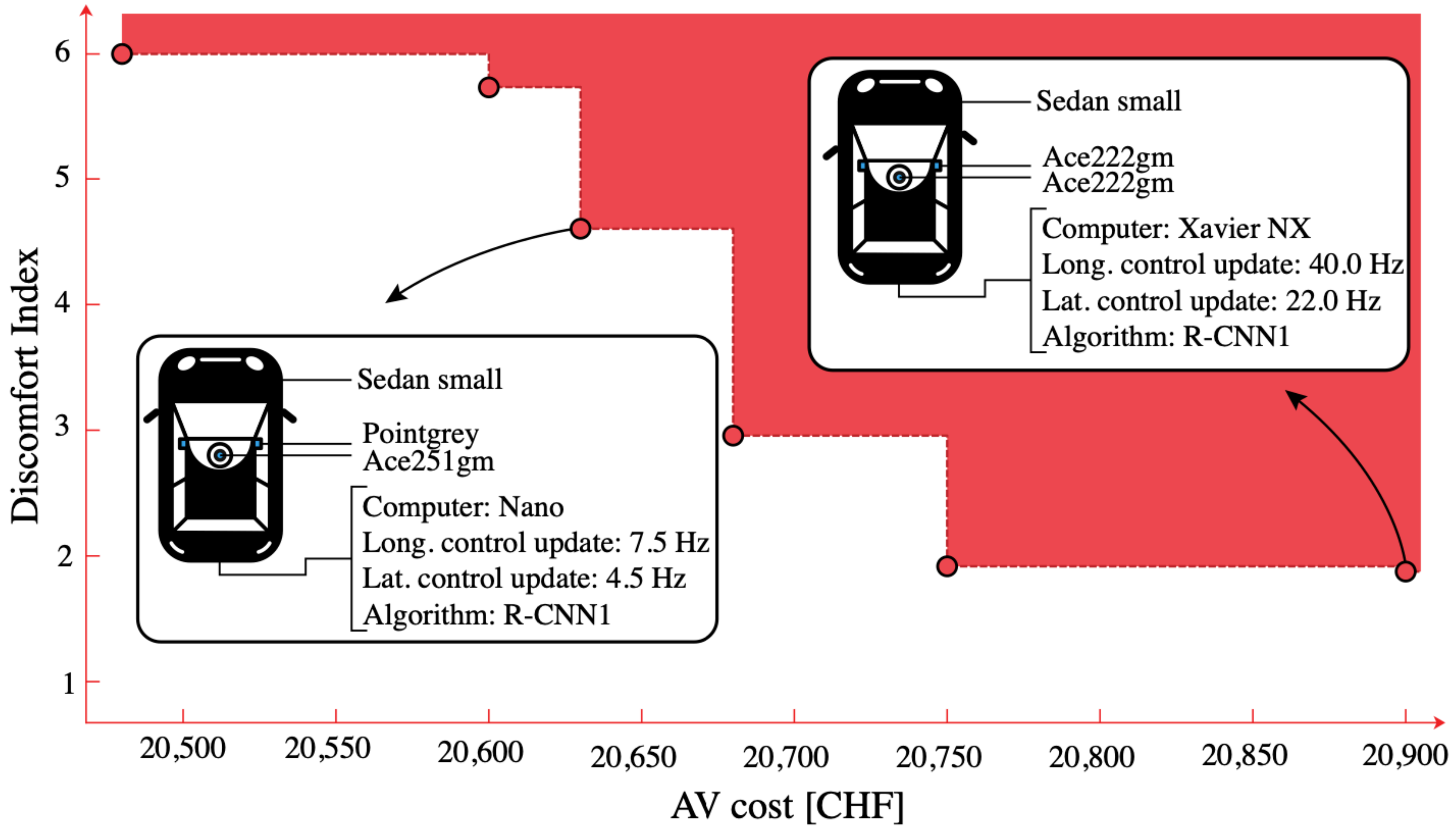
Example: Co-design of embodied intelligence

- **Design trade-offs** via co-design **optimization**

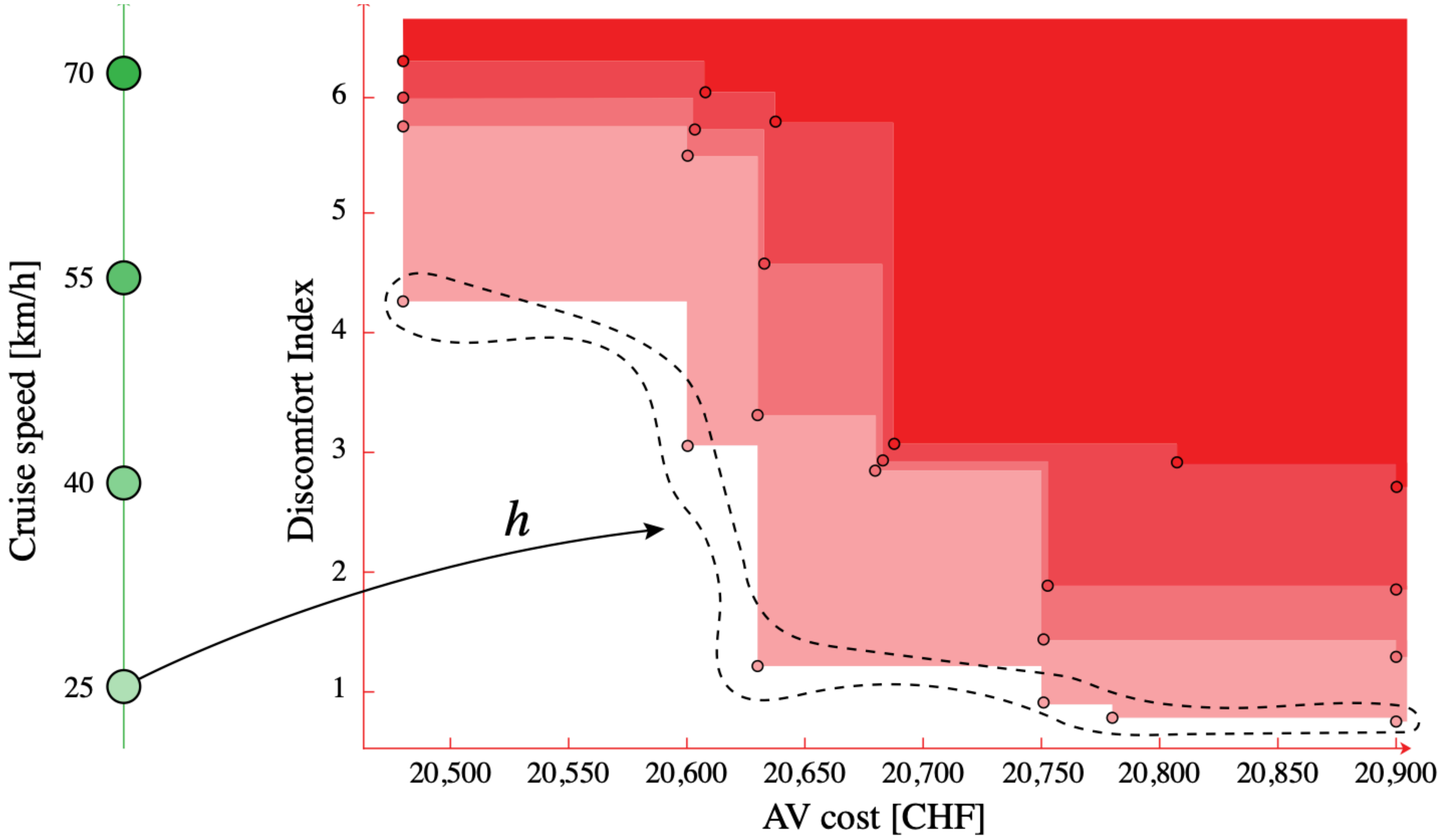


Example: Co-design of embodied intelligence

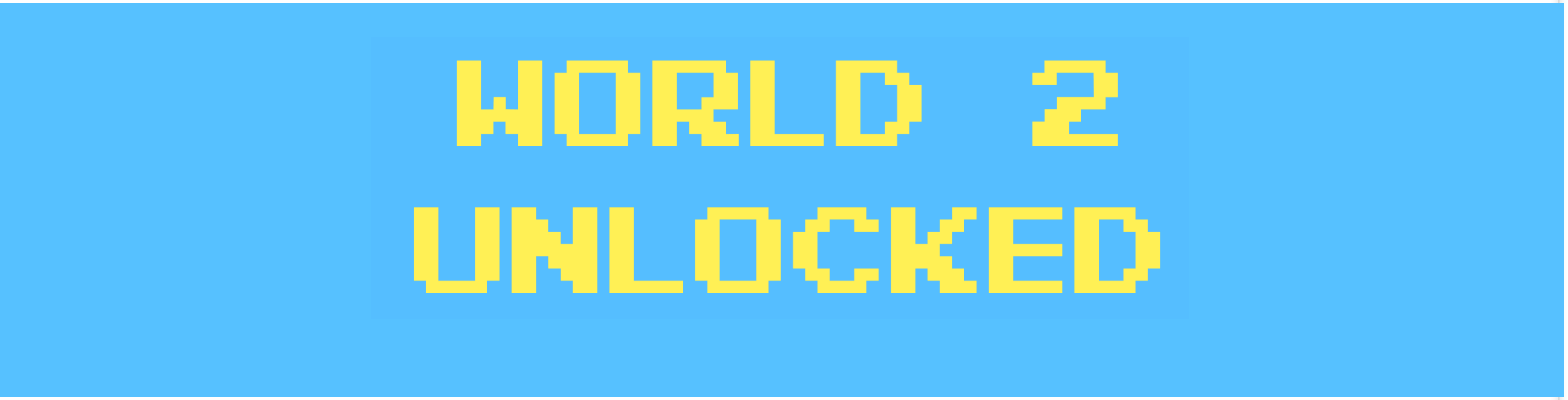
- **Design trade-offs** via co-design **optimization**



- **Monotonicity** at glance



What's next to learn



- Given the things you learned in this class,
  a lot of interesting things are unlocked for you.
- Some of these could be included in an "ACT4E part 2" class.
  Let us know what would interest you in the form.

Where to go to learn more

- Books:
  - Fong, Spivak: An Invitation to Applied Category Theory
  - Spivak: Category Theory for the Sciences
- Conferences/Colloquia/Schools
  - Topos Institute colloquium
  - ACT conference series: ACT 2020 | ACT 2019| ACT 2018
  - The Adjoint School
  - Symposium series: SYCO (Compositional structures)
  - Workshop series: STRING (String Diagrams in Computation, Logic, and Physics)
- Journal: Compositionality
- Chats/forums
  - Stay in the ACT4E Zulip.
  - Category theory Zulip https://categorytheory.zulipchat.com
  - ACT Telegram chat
- Blogs
  - Azimuth blog (John Baez): https://johncarlosbaez.wordpress.com/
  - Algebraic Julia blog: https://www.algebraicjulia.org/blog/
  - Bartosz Milewski: https://bartoszmilewski.com/
  - n-Category Cafe: https://golem.ph.utexas.edu/category/

Computational Trinitarianism

The central dogma of computational trinitarianism holds that ***Logic, Languages, and Categories are but three manifestations of one divine notion of computation****.*
There is no preferred route to enlightenment: each aspect provides insights that comprise the experience of computation in our lives.

Computational trinitarianism entails that any concept arising in one aspect should have meaning from the perspective of the other two. If you arrive at an insight that has importance for logic, languages, and categories, then you may feel sure that you have elucidated an essential concept of computation–you have made an enduring scientific discovery. (Harper)

- https://ncatlab.org/nlab/show/computational+trinitarianism

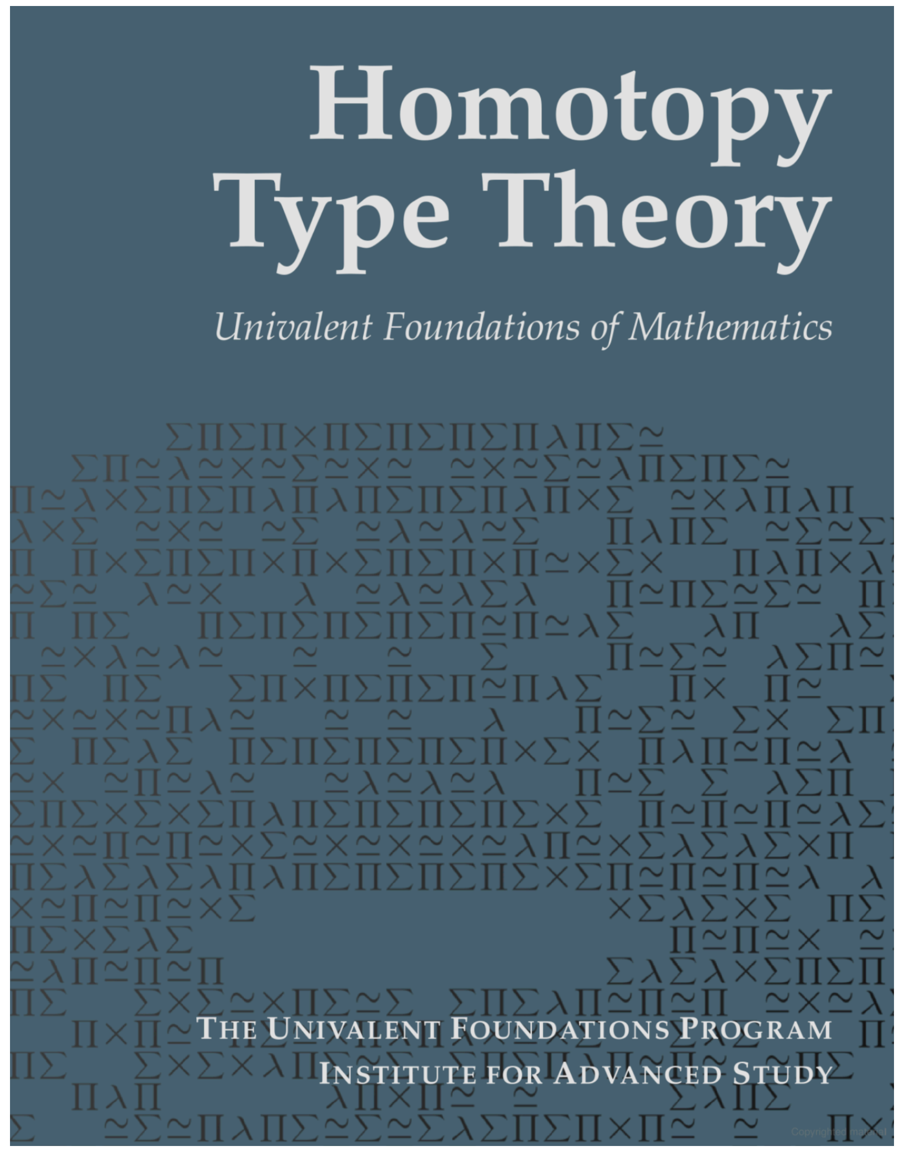
- Baez, Stay *Physics, Topology, Logic and Computation: A Rosetta Stone*

Category Theory	Physics	Topology	Logic	Computation
object X	Hilbert space X	manifold X	proposition X	data type X
morphism $f\colon X \to Y$	operator $f\colon X \to Y$	cobordism $f\colon X \to Y$	proof $f\colon X \to Y$	program $f\colon X \to Y$
tensor product of objects: $X \otimes Y$	Hilbert space of joint system: $X \otimes Y$	disjoint union of manifolds: $X \otimes Y$	conjunction of propositions: $X \otimes Y$	product of data types: $X \otimes Y$
tensor product of morphisms: $f \otimes g$	parallel processes: $f \otimes g$	disjoint union of cobordisms: $f \otimes g$	proofs carried out in parallel: $f \otimes g$	programs executing in parallel: $f \otimes g$
internal hom: $X \multimap Y$	Hilbert space of 'anti-X and Y': $X^* \otimes Y$	disjoint union of orientation-reversed X and Y: $X^* \otimes Y$	conditional proposition: $X \multimap Y$	function type: $X \multimap Y$

Table 4: The Rosetta Stone (larger version)

Homotopy Type Theory

- The basics of Homotopy Type Theory are a good introduction to the **mechanization of mathematics**.



- Example: how to define a partial order?

partial order

reflexive	*transitive*	*antisymmetric*
$\dfrac{\top}{aRa}$	$\dfrac{aRb \quad bRc}{aRc}$	$\dfrac{aRb \quad bRa}{a = b}$

- In this interpretation:

aRb **type of proofs** that a relates to b

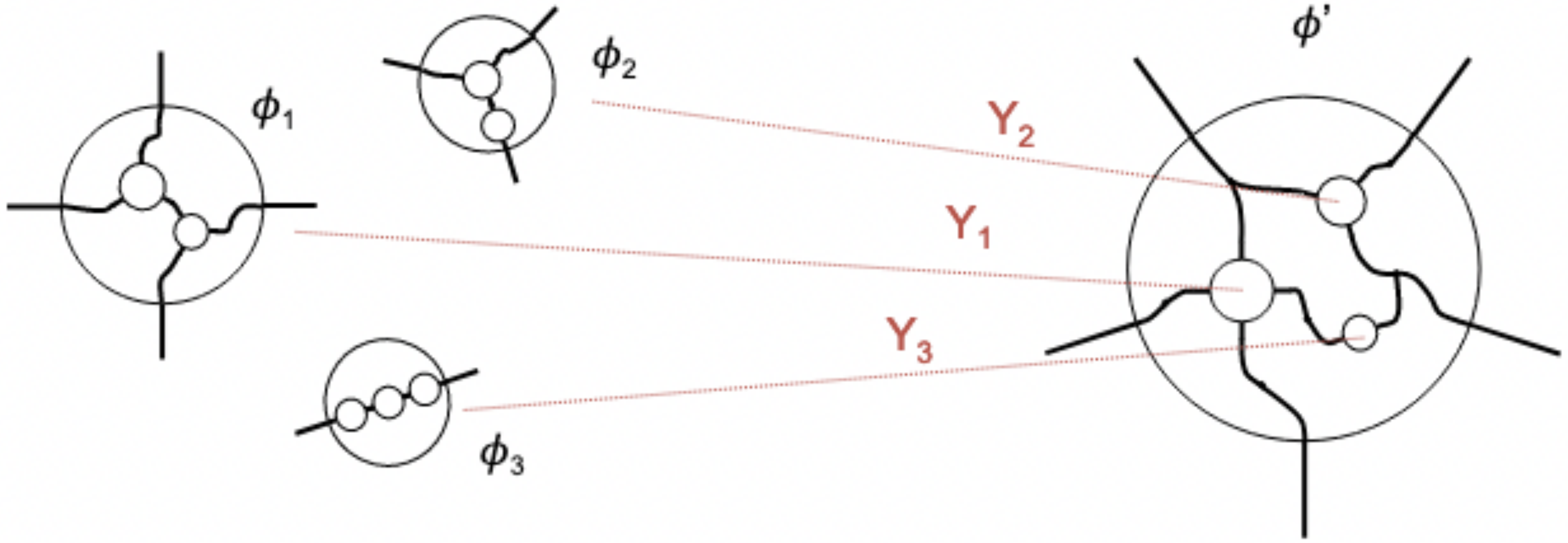
$a = b$ **type of proofs** that a is equal to b

- Defining a poset means providing 3 functions:

$$\begin{aligned} \mathsf{reflexive} &: (a : X) \to (aRa) \\ \mathsf{transitive} &: (a : X) \times (b : X) \times (c : X) \times (p_1 : aRb) \times (p_2 : bRc) \to (aRc) \\ \mathsf{antisymmetric} &: (a : X) \times (b : X) \times (p_1 : aRb) \times (p_2 : bRa) \to (a = b) \end{aligned}$$

Operads, wiring diagrams

- Operads look at **a higher level of compositionality** (compositionality of compositions).

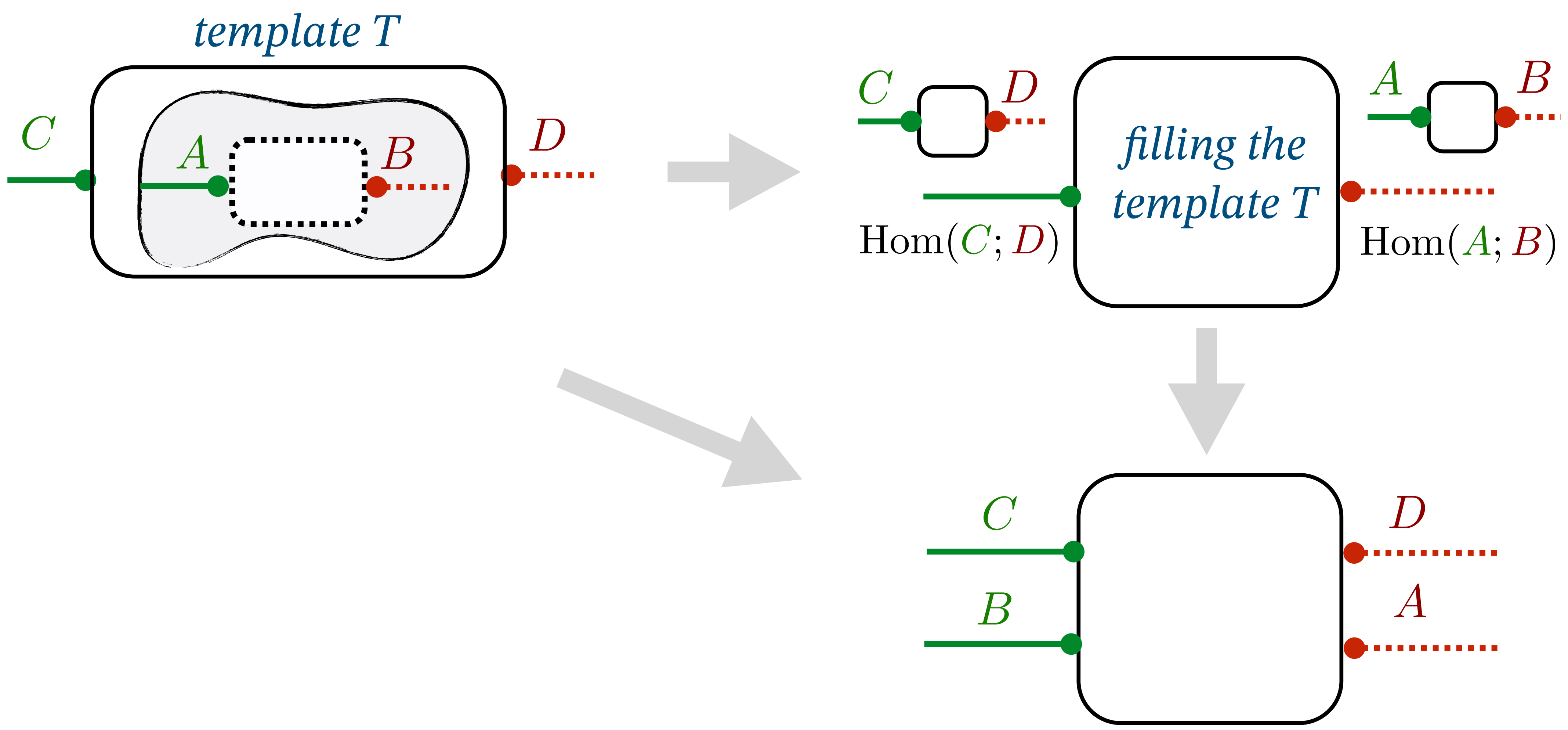


D. Spivak - Operads of wiring diagrams

- Resources
  - Tai-Danae Bradley: https://www.math3ma.com/blog/what-is-an-operad-part-1
  - D. Spivak: Category Theory for the Sciences

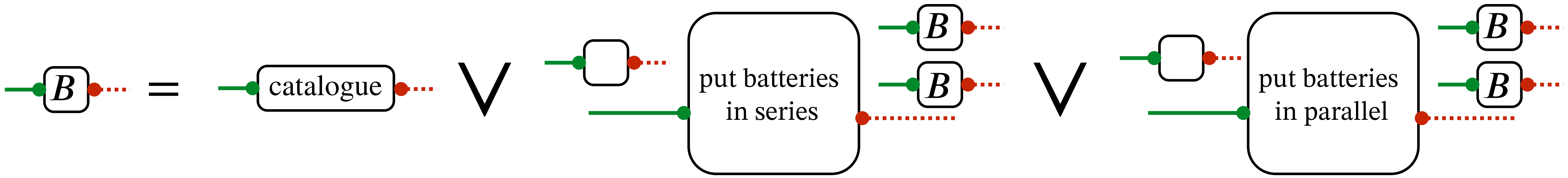
Operads, wiring diagrams

- Higher-order structure, or lack thereof.



- Semantics of **recursion in DP**
  - I can make a battery by putting two batteries in parallel or in series.

$$\text{battery} = \text{catalogue} \vee \text{series}(\text{battery}, \text{battery}) \vee \text{parallel}(\text{battery}, \text{battery})$$



Linear logic

- Linear logic is a "**logic of resources**".

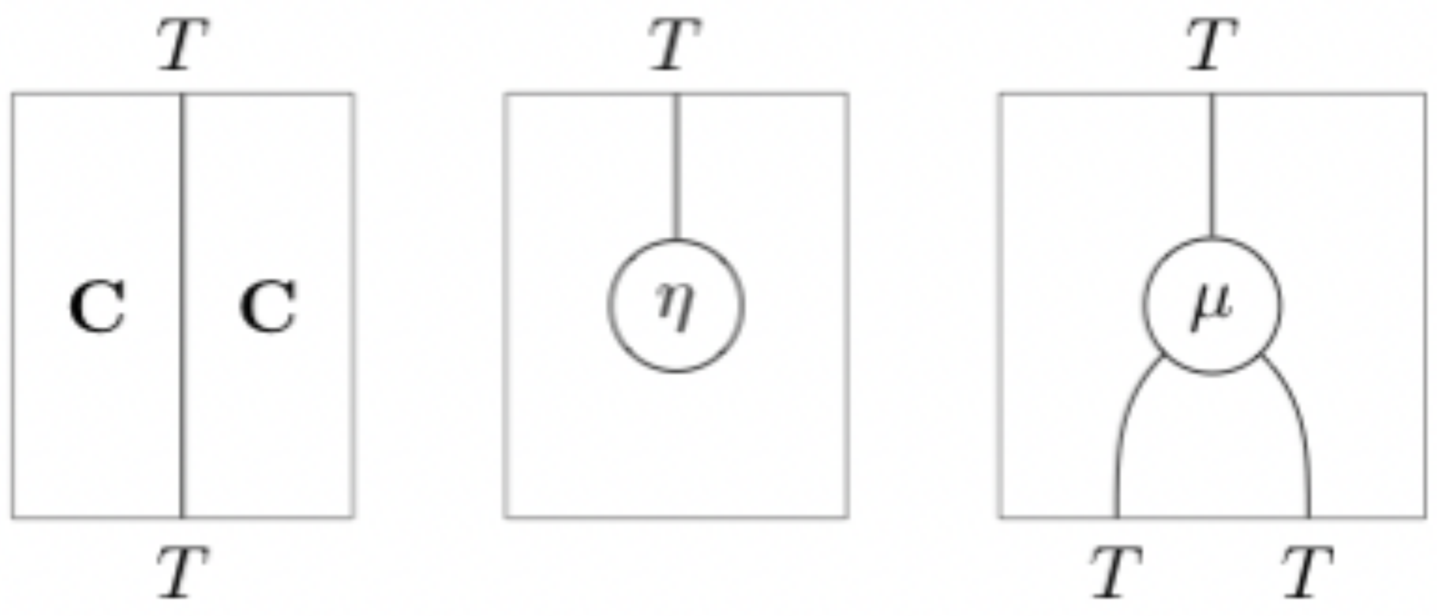
- Two "multiplicatives", two "additives", four units, an involution, two modalities!

		unit
$A \mathbin{\&} B$	you can choose to have either an A or a B	$\top$
$A \oplus B$	somebody chooses if you have an A or a B	$\mathbf{0}$
$A \otimes B$	you have both an A and a B	$\mathbf{1}$
$A \parr B$	if you understand this, you are enlightened	$\perp$
$!A$	you have 0 or more of A	
$?A$	you can discard 0 or more of A	
A	you have an A	
$A^{\perp}$	you have a debt of one A	

- Compared to co-design:
  - only boolean resources
  - richer conjunctions
  - cannot split wires or terminate wires.

Monads and comonads

- String diagrams representation



- Monads as the extra structure that you need to create categories of "generalized functions" (Kleisli categories):

  "A monad is a consistent way of extending spaces to include generalized elements and generalized functions of a specific kind." - P. Perrone

$$f : X \to Y$$

$$F : X \to MY \quad \begin{cases} \text{power set of Y} \\ \text{probability distributions on Y} \\ \text{intervals of Y} \\ \text{sequences of Y} \end{cases}$$

Monads and comonads

- Categories of algebras (Eilenberg-Moore category of a Monad)

  "A monad is a consistent choice of formal expressions of a specific kind, together with ways to evaluate them." - P. Perrone

  Monoids = algebras for the list Monad

  Groups, Rings, Vectors spaces

  Lattices, Pointed sets

  Etc.

$$
\begin{tikzcd}
A \arrow[r, "\eta"] \arrow[dr, "\mathrm{id}"'] & TA \arrow[d, "e"] \\
 & A
\end{tikzcd}
\qquad
\begin{tikzcd}
TTA \arrow[r, "Te"] \arrow[d, "\mu"] & TA \arrow[d, "e"] \\
TA \arrow[r, "e"] & A
\end{tikzcd}
$$

- Comonads (+ co-Kleisli morphisms and coalgebras)

  "A comonad is a consistent way to construct, from spaces, processes of a specified structure, and give selected strategies or trajectories" - P. Perrone

- Resources
  1. P. Perrone: Notes on category theory with examples from basic mathematics
  2. E. Riehl: Category Theory in Context
  3. Bartosz Milewski's Youtube lectures

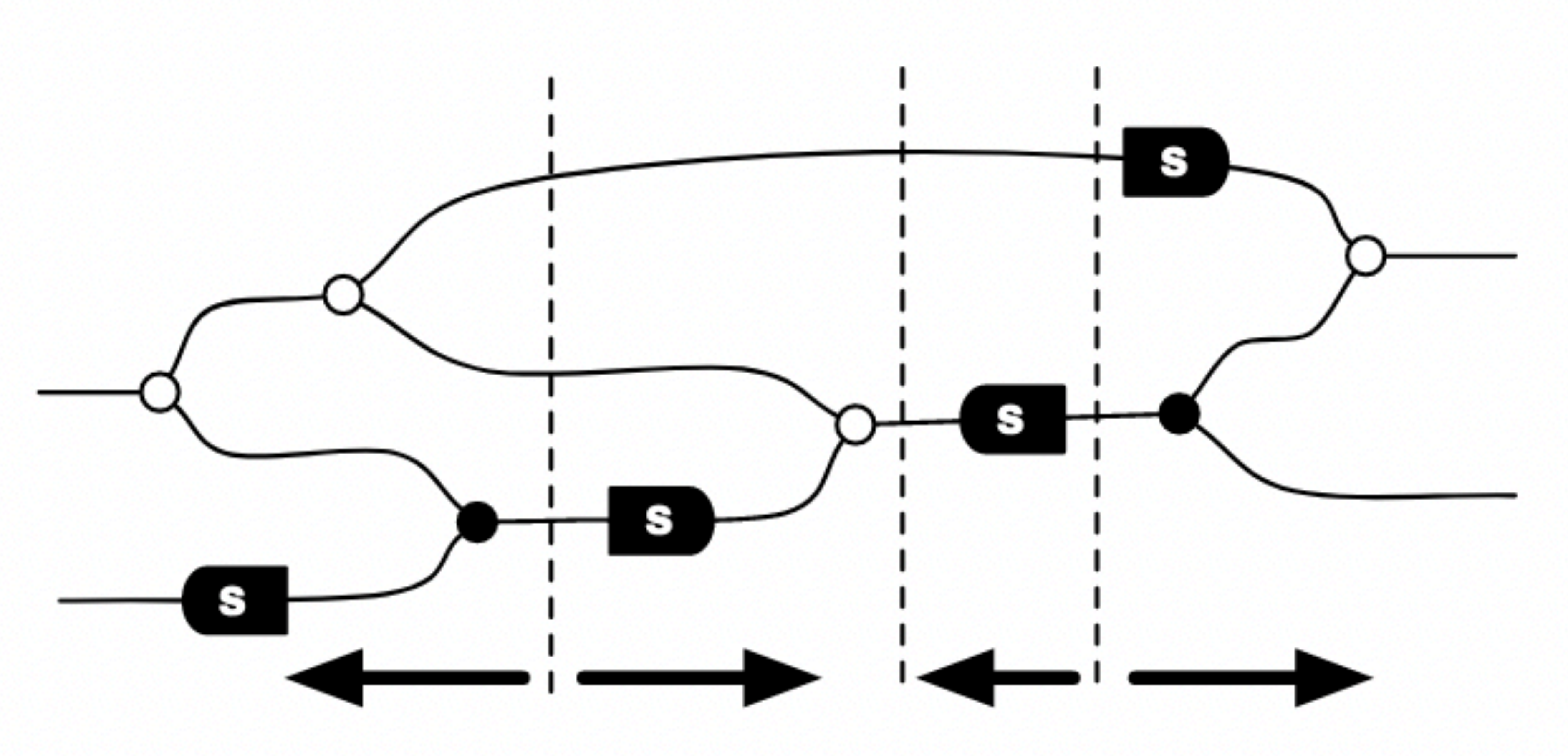
Props

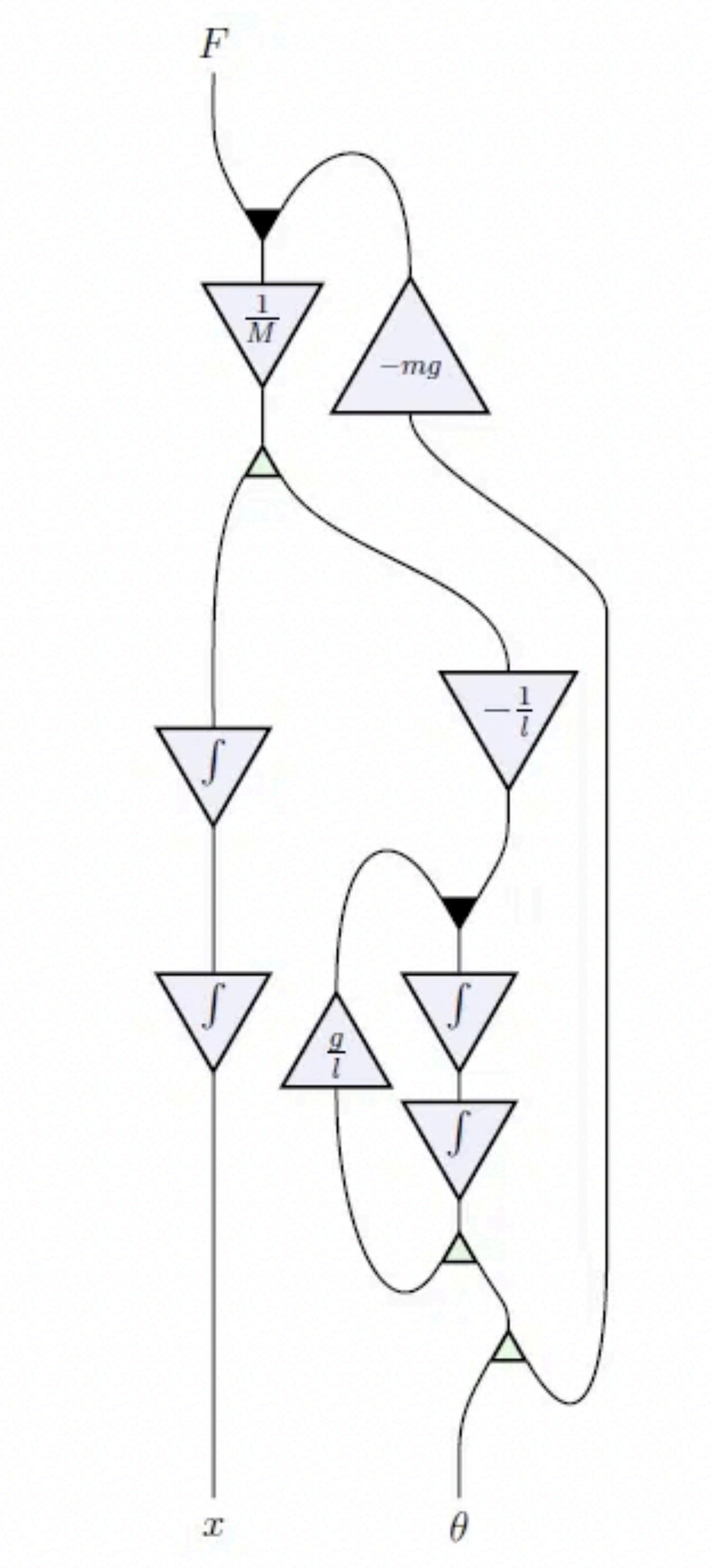
- Syntax for algebraic structures
  - Network theory

    Signal flow diagrams

    Circuit-style diagrams
  - Graphical linear algebra
  - Dynamical systems

$$x^{\otimes n} = x \otimes x \otimes \cdots \otimes x$$

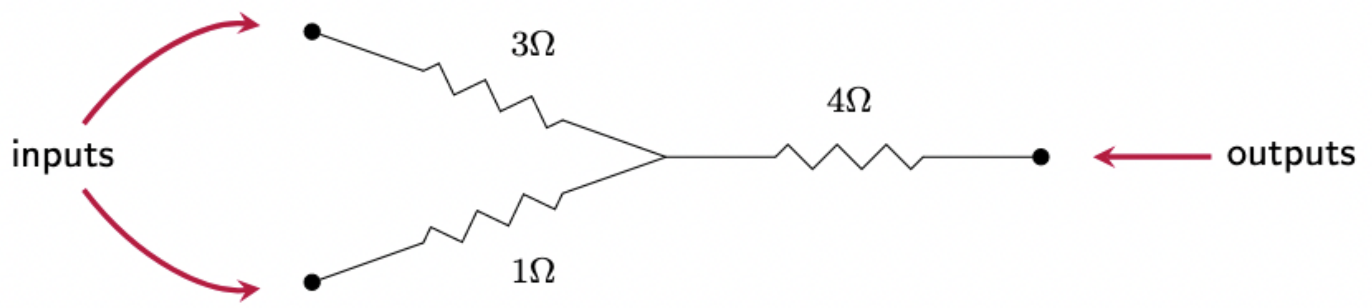


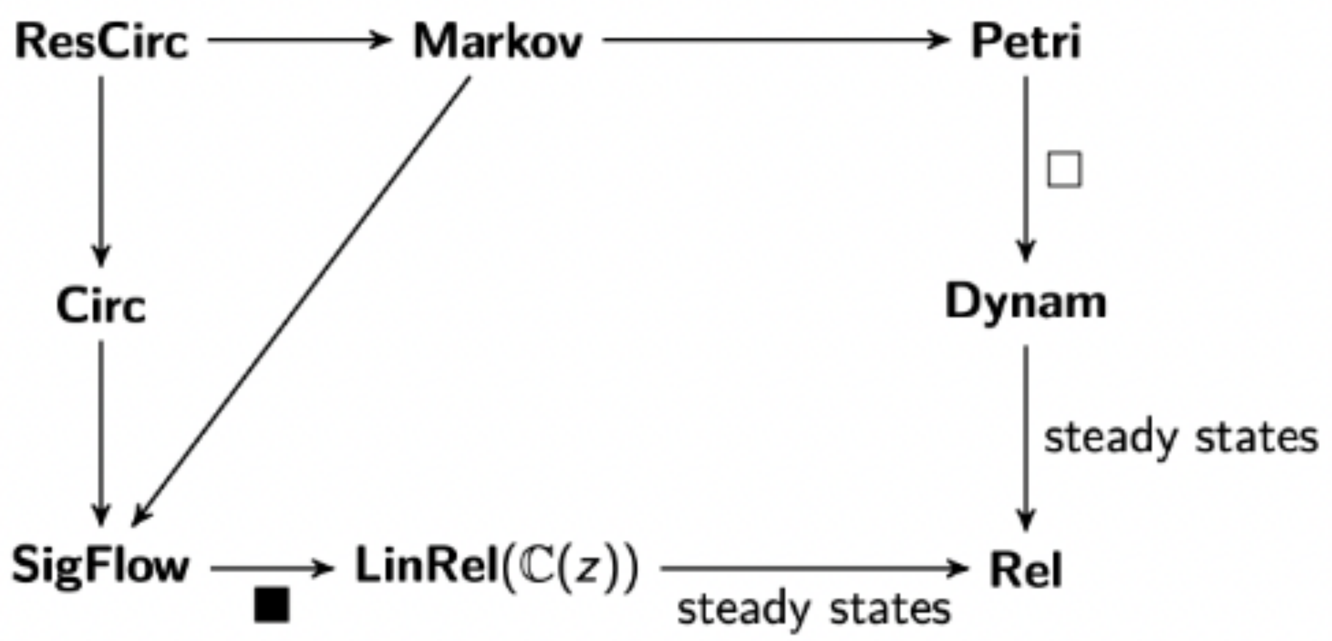


- Resources:
  1. https://johncarlosbaez.wordpress.com/2018/04/27/props-in-network-theory/
  2. https://graphicallinearalgebra.net/
  3. Fong, Sobocinski, Rapisarda, A categorical approach to open and interconnected dynamical systems

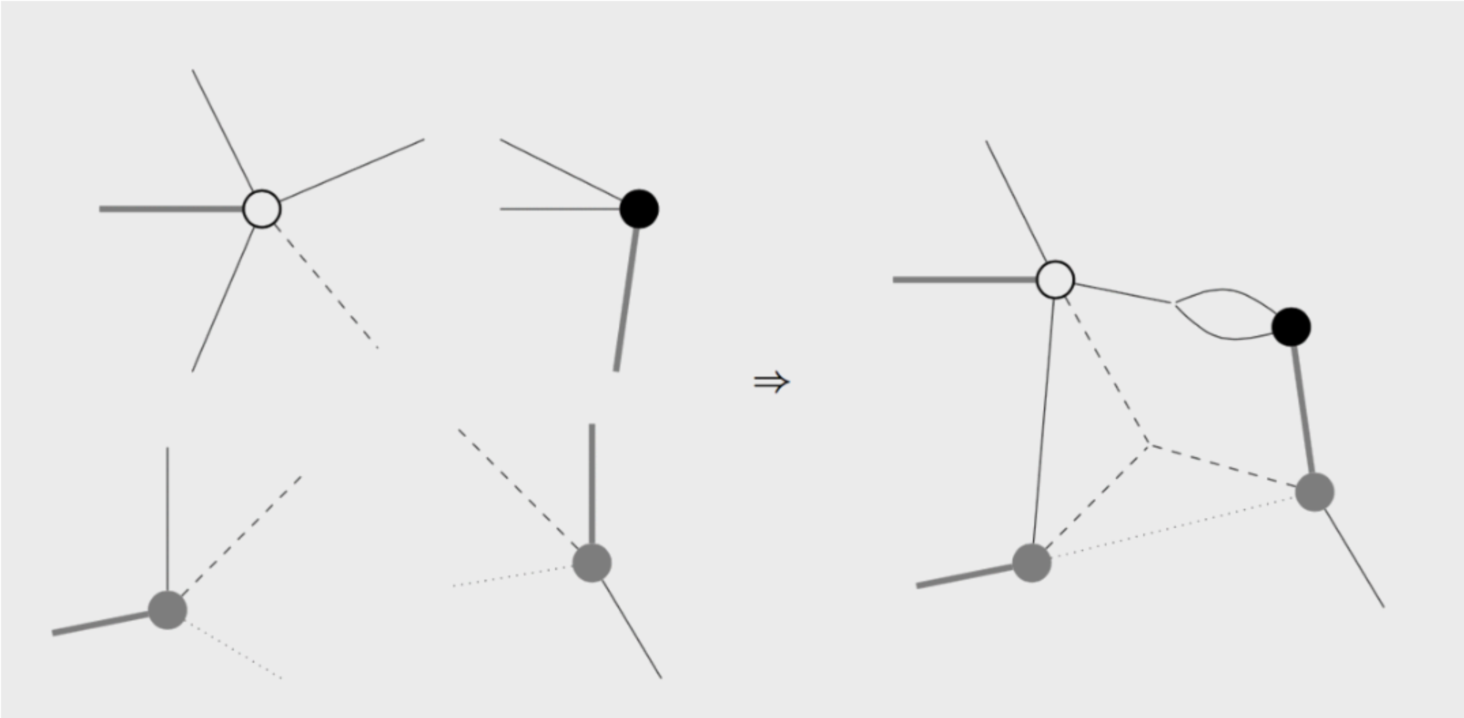
Decorated/structured cospans

- Open networks





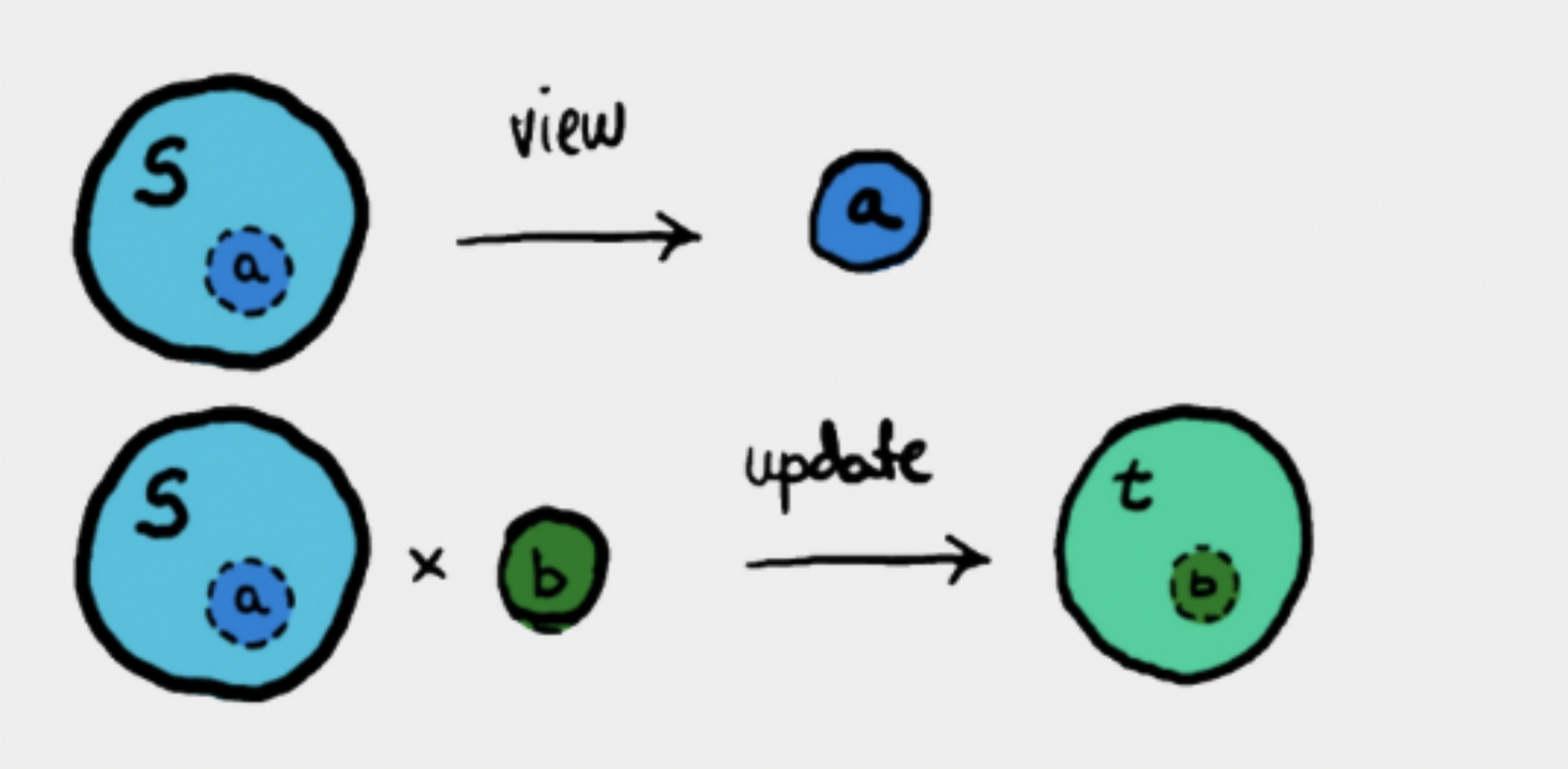
- Hypergraph categories



- Blog posts, with further references:
  1. https://johncarlosbaez.wordpress.com/2015/05/01/decorated-cospans/
  2. n-Category café post on hypergraph categories of cospans
  3. https://johncarlosbaez.wordpress.com/2021/01/30/structured-vs-decorated-cospans/
  4. Baez, Courser: Open systems, a double-categorical perspective

Optics

- Lenses

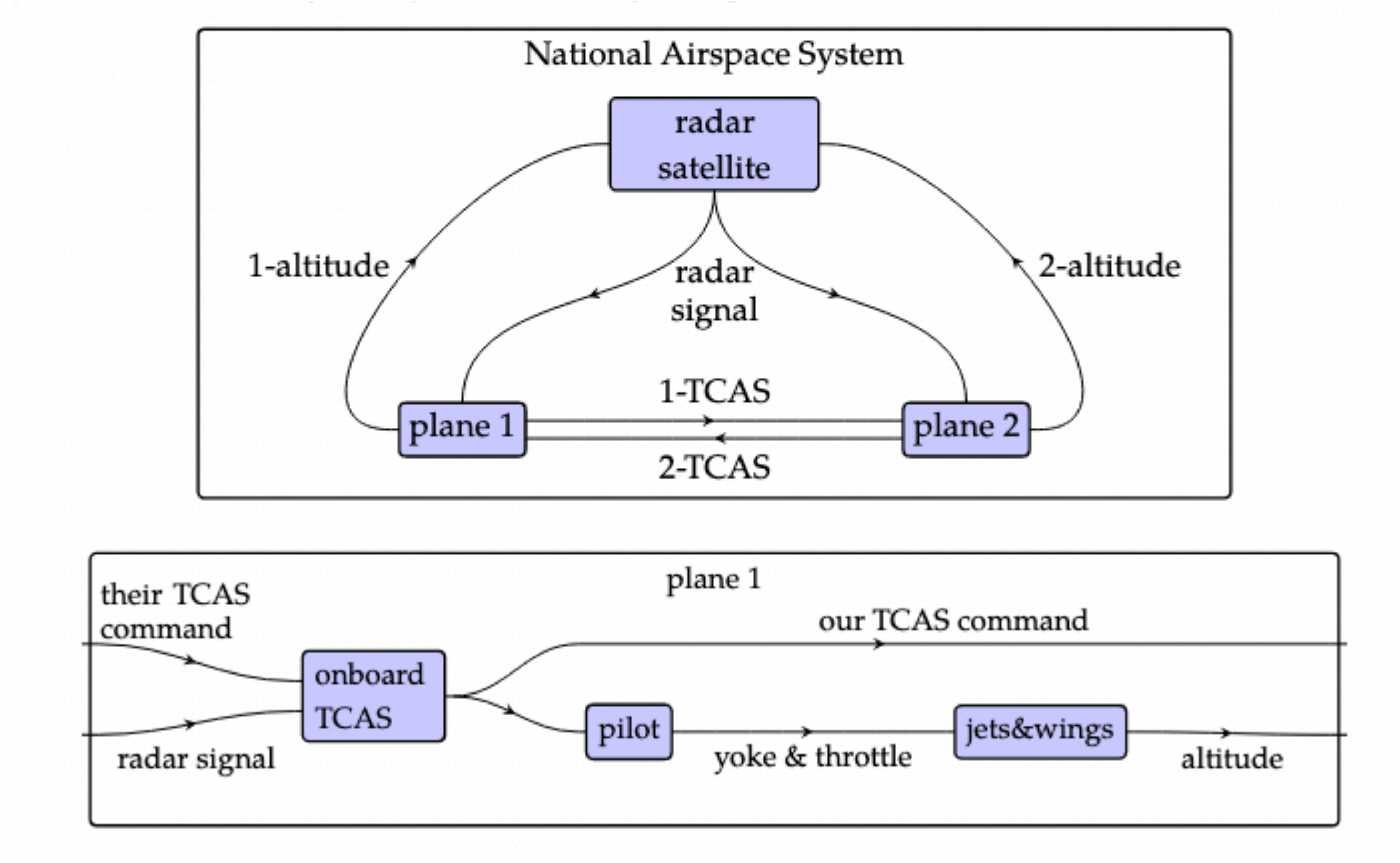


Emily Pillmore and Mario Román - Profunctor optics

- Resources
  1. n-Category café post on Profunctor optics
  2. Fong, Johnson: *Lenses and learners* https://arxiv.org/abs/1903.03671
  3. Jules Hedges on lenses and open games

Sheaves, Toposes, Temporal logic

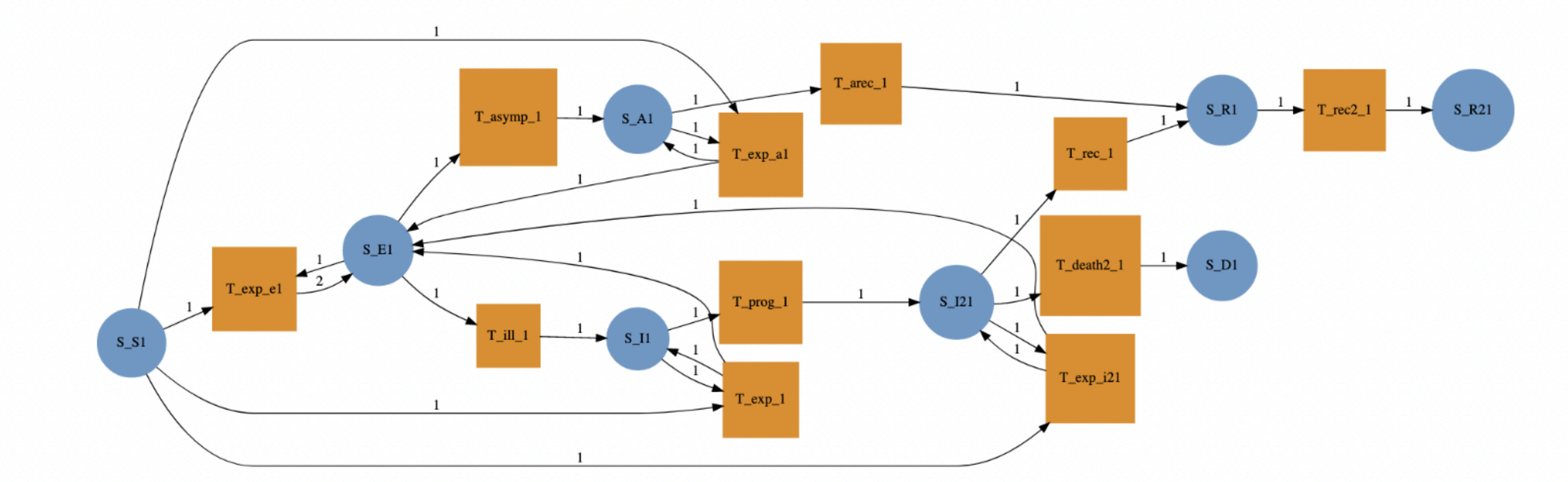
- Generalized dynamical systems: behaviors in time
  - How can we prove our machine/system is safe?
  - Is there a formal logic to describe system behaviors over time?



- Resources
  1. Chapter 7 in An Invitation to Applied Category Theory
  2. Schultz, Spivak, Vasilakopoulou: Dynamical Systems and Sheaves: https://arxiv.org/abs/1609.08086
  3. Schultz, Spivak: Temporal Type Theory: https://arxiv.org/abs/1710.10258
  4. Zardini, Spivak, Censi, Frazzoli: A Compositional Sheaf-Theoretic Framework for Event-Based Systems

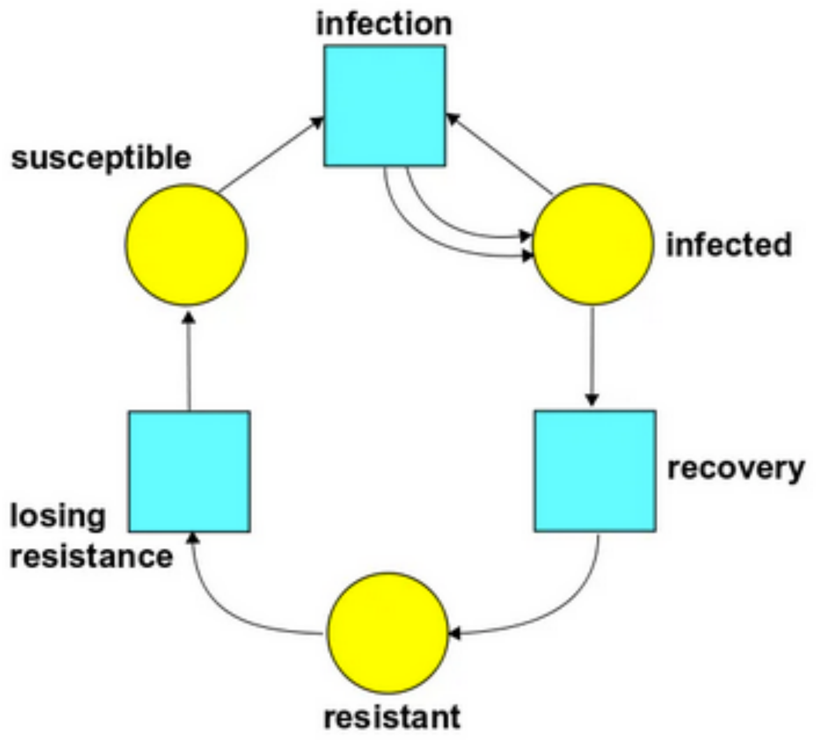
Petri nets

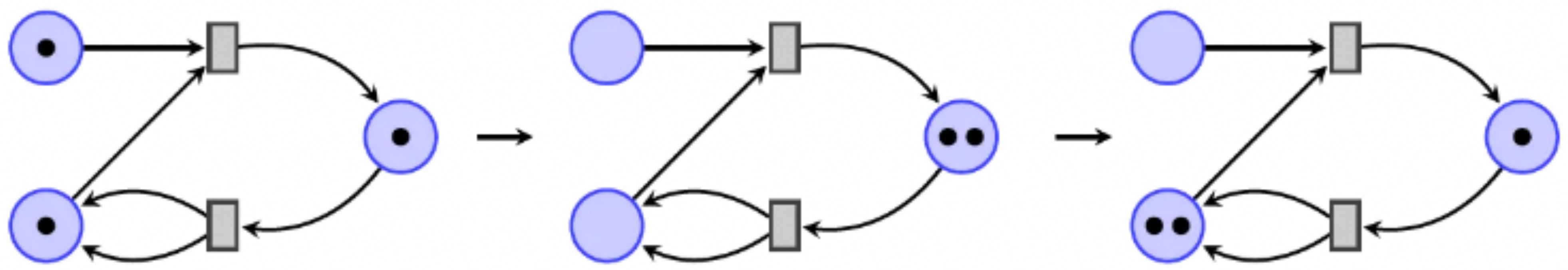
- Petri nets



Compositional epidemiological modeling using structured cospans

Micah Halter and Evan Patterson



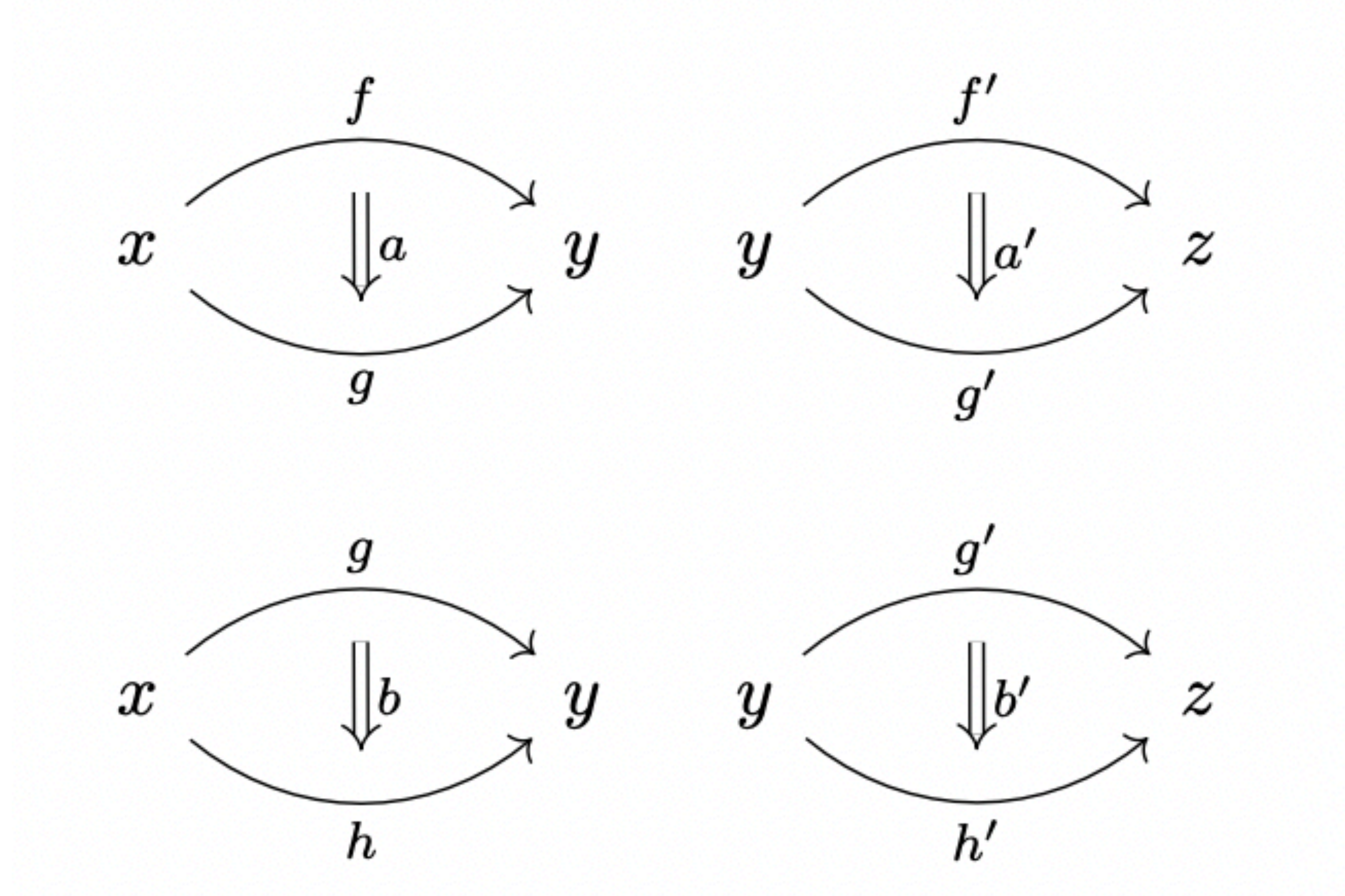


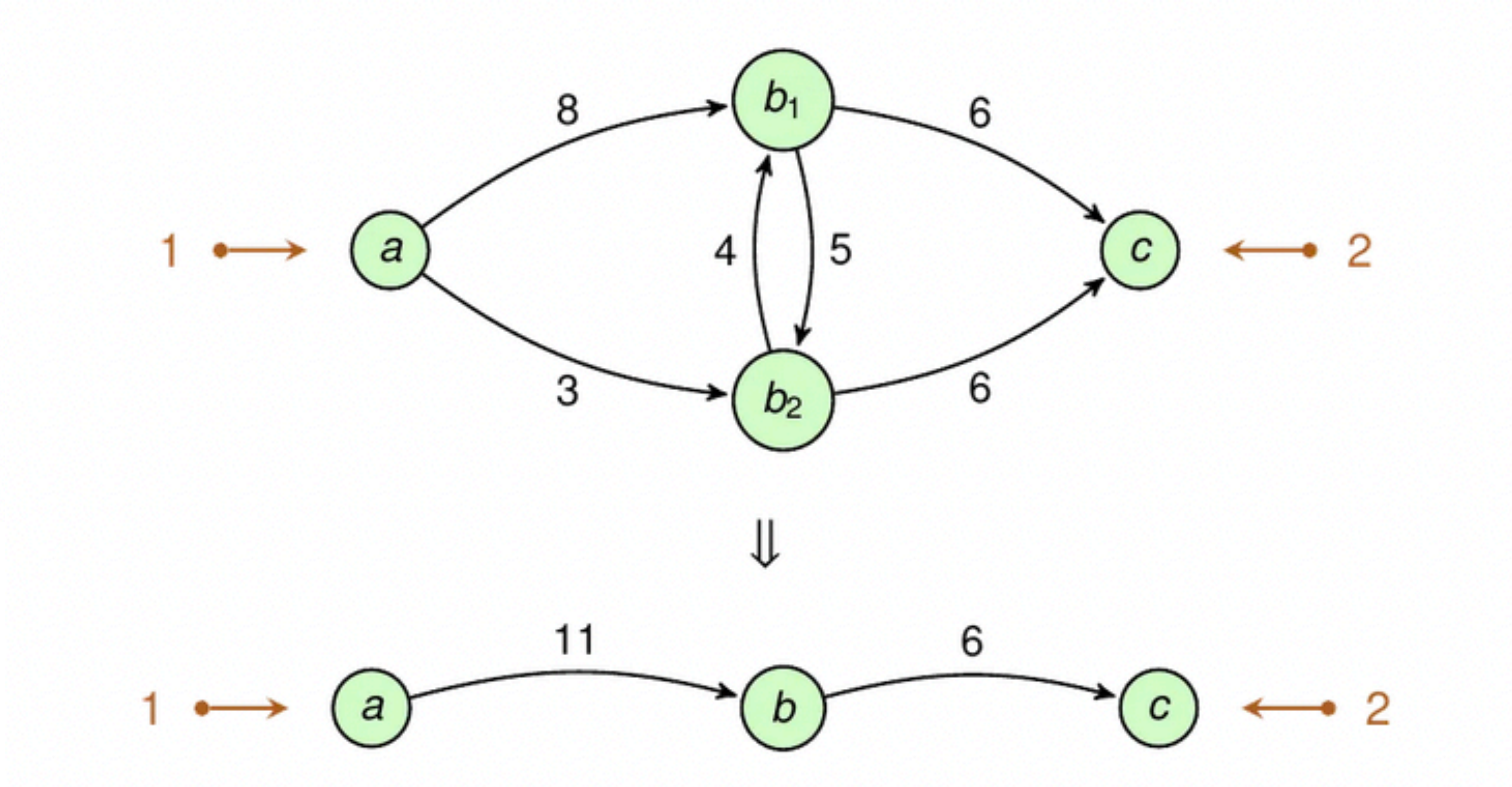
John C. Baez, Fabrizio Genovese, Jade Master, Mike Shulman

- Resources

1. https://www.algebraicjulia.org/blog/post/2020/10/structured-cospans
2. Statebox: What is statebox? | A Library to do Category Theory in Idris
3. https://johncarlosbaez.wordpress.com/2021/01/17/categories-of-nets/

Bicategories and more





- Bicategories
- Double-categories
- Enriched category theory
- Structured categories of various flavors
  - *-autonomous categories
  - Closed categories
  - Monoidal categories with additional algebraic structures
  - etc.
- Polycategories, Multicategories, …

Next steps

- We need to **finish the part of the book** that includes the course materials.

- We are going to write up in-depth **case studies**:
  - Restate **control theory** in category theory
  - Co-design perspective on **embedded systems** (optimize software+HW together)
  - ...

- Between Feb 22nd - June 4th we teach an extended version of this class at ETH.
  - We will **extend the materials** we have to cover some of the next topics mentioned.
  - We will add **code exercises.**

- **Keep working with us:**
  - help with the book
  - help with the code
  - beta-test our new materials and exercises
  - suggest new case studies

Feedback form

- You will receive an anonymous **feedback form**
  - we'd love to hear your impressions, comments, and suggestions!