Designing Everything (,) Together

Resources and co-design theories
for formal engineering design.

- **Andrea Censi**
  - Senior researcher at ETH Zürich
  - President, Duckietown Foundation
  - Founder, Züpermind

Resource accounting was an early motivation for math (1)

- **You are a sheep herder in 5,000 BCE,** and you have “a lot” of sheep which you take out grazing every day.
- How can you make sure at the end of the day that no sheep was lost?

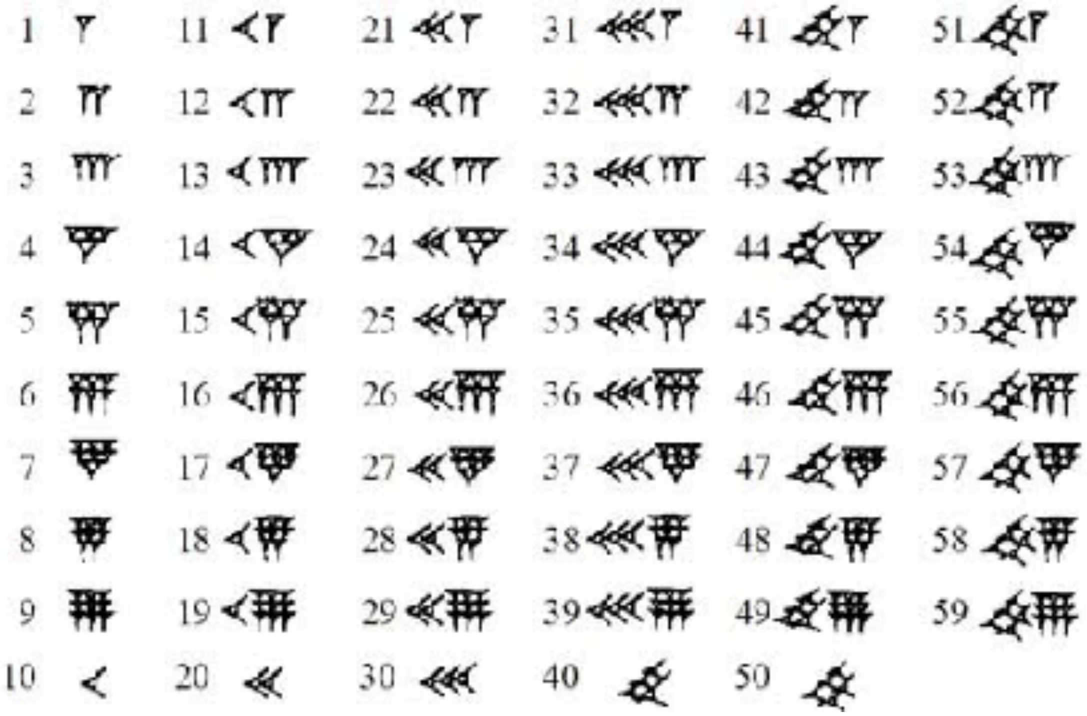
- **The rope method**
  - Take a rope.
  - Tie one knot for every sheep that exits the gate.
  - Untie one knot for each sheep that you bring back at night. (You know the name of each sheep, but it does not matter which sheep)
  - If you have knots left over: go look for the missing sheep!

- **Can you read between the lines?**
  - Definition of **natural numbers**
  - Definition of **cardinality** of a set, isomorphisms between sets.

Resource accounting was an early motivation for math (2)

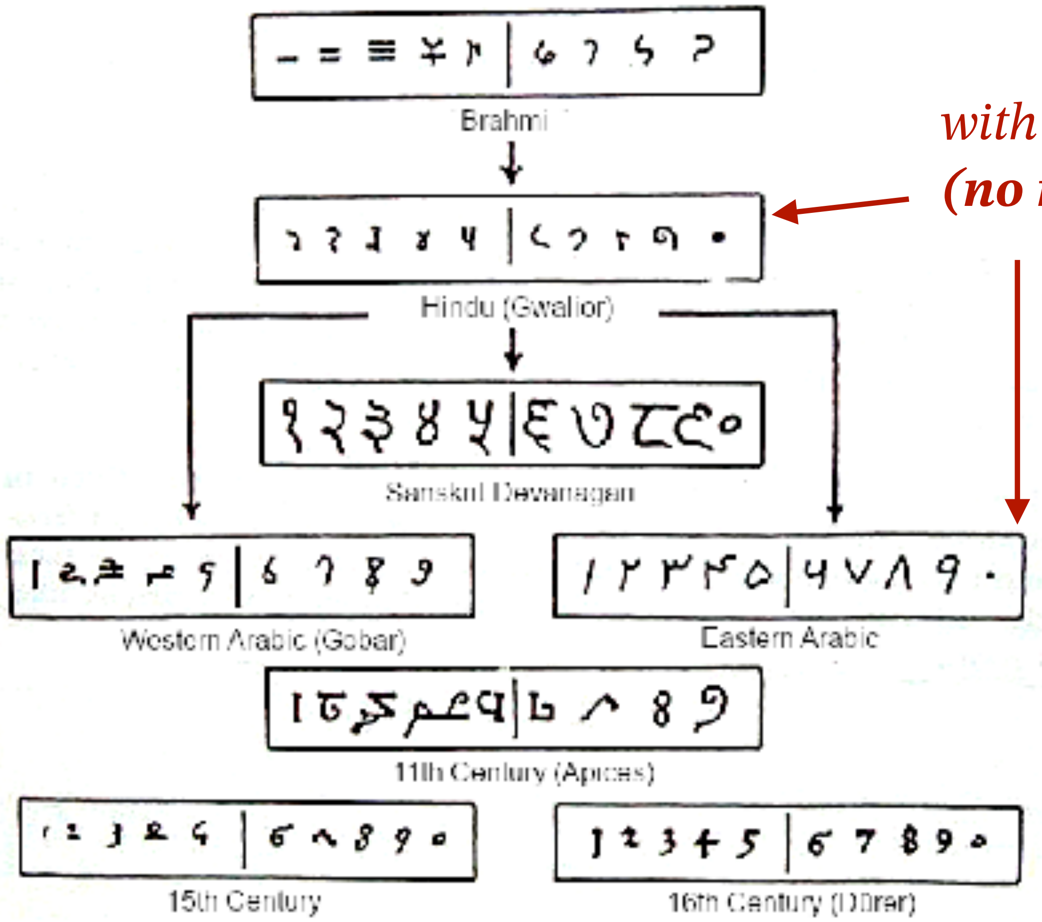
- You are **a heir to a sheep empire in 3,000 BCE Babylon.** You have a lot of sheep and other riches to count!
- **Abstractions of resources** like **number systems** allows to keep track of resources in a **compositional** way.

3,000 BCE, Babylonian positional number system



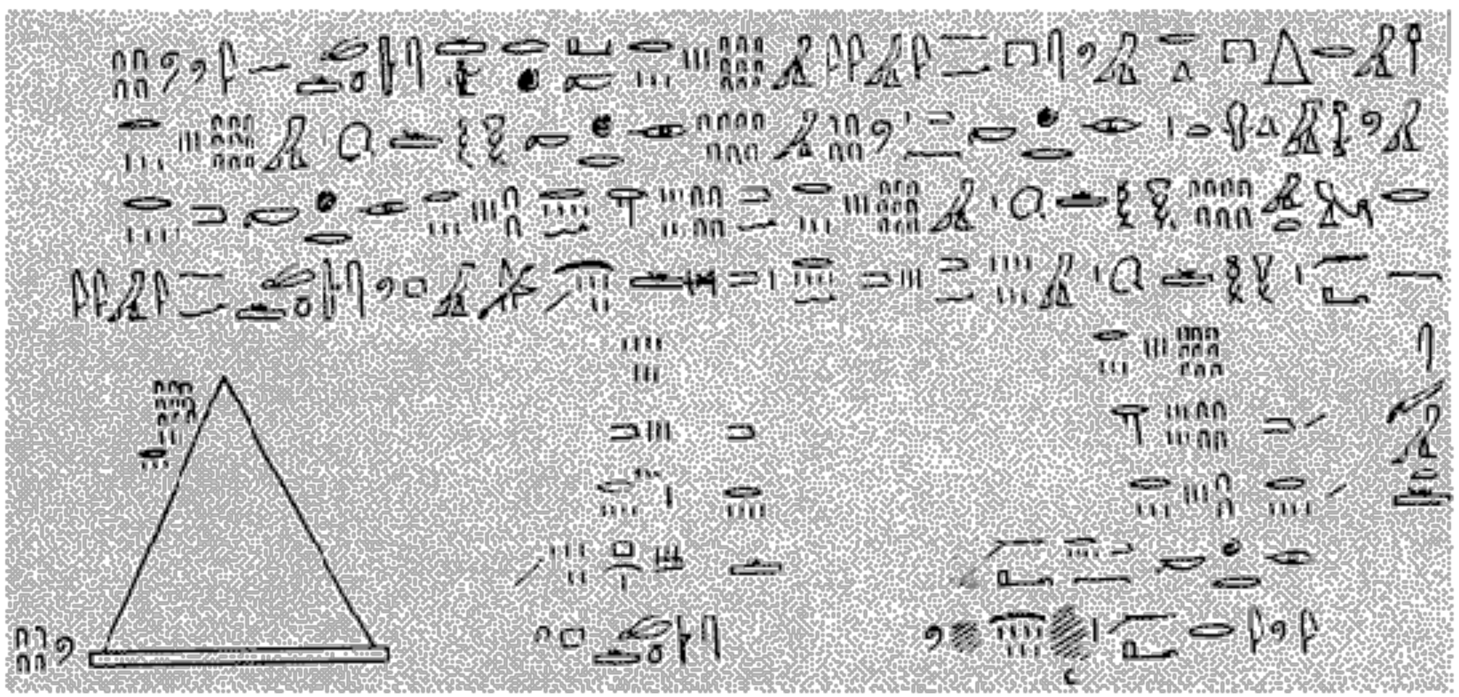
evolution of Hindu-Arabic modern number systems

with zeros ***(no resources)***

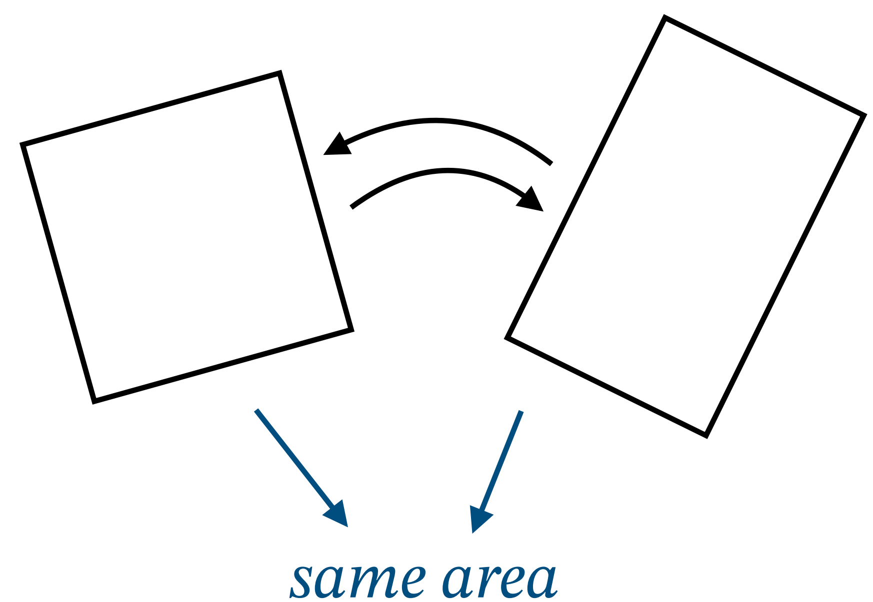


Resource accounting was an early motivation for math (3)

- You are **an Egyptian scribe in 2,000 BCE.**
- How can you **divide the land resources fairly** after the annual flood erased last year's boundaries?



- How can you change one shape into another?
  - **rigid transformations**
  - **joining, diving**
- You recognize equivalence through **invariants preserved by the operations.**

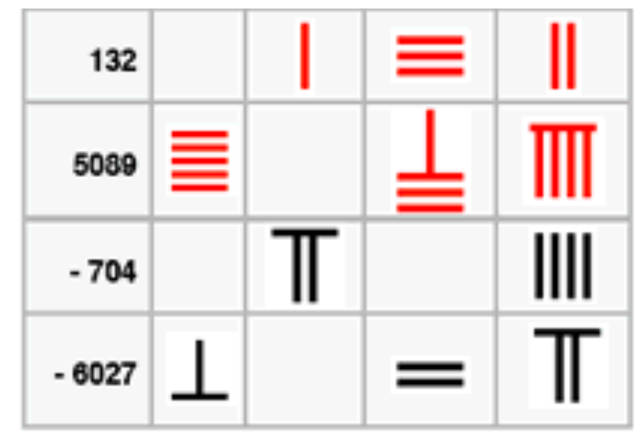


Resource accounting was an early motivation for math (4)

- You are a **merchant in India in 600 CE**. You need to deal with taxes, loans, and other financial instruments.
- You need the notion of **negative resources.**

Brahmagupta, India, 620 CE

Accounting systems China, 200 BCE



*A **debt minus zero** is a **debt**.*
*A **fortune minus zero** is a **fortune**.*
*A **zero minus zero** is a **zero**.*
*A **debt subtracted from zero** is a **fortune**.*
*A **fortune subtracted from zero** is a **debt**.*
*The **product** of **zero** multiplied by a **debt** or **fortune** is **zero**.*
*The **product** of **zero** multiplied by **zero** is **zero**.*
*The **product or quotient** of two **fortunes** is one **fortune**.*
*The **product or quotient** of two **debts** is one **fortune**.*
*The **product or quotient** of a **debt** and a **fortune** is a **debt**.*
*The **product or quotient** of a **fortune** and a **debt** is a **debt**.*

- Operations and identities, monoids, conjugations...
- Homomorphism from numbers to {**debt**, **zero**, **fortune**}.

And then?

- Having established proper accounting of sheep, land, and taxes, **math decides to grow beyond thinking about resources**.
  - Friends say she had a fallout with Philosophy and Engineering.
  - Reportedly she went in search of "truth" and "beauty".
  - Still had occasional flings with the real world with Physics.

To the 21st century!

22nd Century Problems

- You are a sentient AI in 2125 trying to bootstrap to singularity before the humans notice.

21st Century problems

- You are **a few hundreds engineers in 2020** who need to **design a fleet of self-driving cars**.

The pain of engineering complex systems

a fleet of self-driving cars =

software
behavior
coordination
hardware
actuation
localization
planning
social acceptance
sensing
computation
control
perception
learning
liability
energetics
communication
mapping
regulations
interaction

So many components (hardware, software, ...), so many choices to make!

Nobody can understand the whole thing!

We forget why we made some choices, and we are afraid to make changes...

These "computer" thingies are not helping us that much for design...

"My dear, it's simple: you lack a proper theory of co-design!"

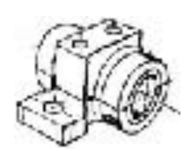
anthropomorphization of 21st century engineering malaise

Some references

- Book
  - Fong, Spivak: *Seven Sketches in Compositionality*, Chapter 4.
- Papers
  - Censi. *A Mathematical Theory of Co-Design 2015*
  - Censi. *Uncertainty in monotone co-design problems*. 2017
  - Zardini et al. *Towards a Co-Design Framework for Future Mobility Systems* 2019
- Online classes
  - Fong, Spivak - An Invitation to Applied Category Theory
    http://math.mit.edu/~dspivak/teaching/sp18/
  - Same material: Applied Category Theory by John Baez (Azimuth)

- A **design problem** is a **relation** between **provided functionality** and **needed requirements.**

design
problem



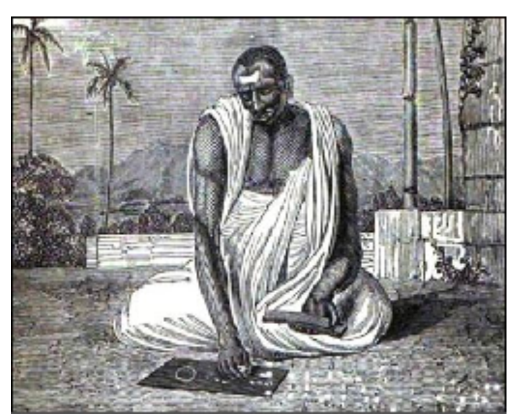
provided
functionality

needed
requirements

“functional requirements”
“desired behavior”
“required performance”
“specifications”

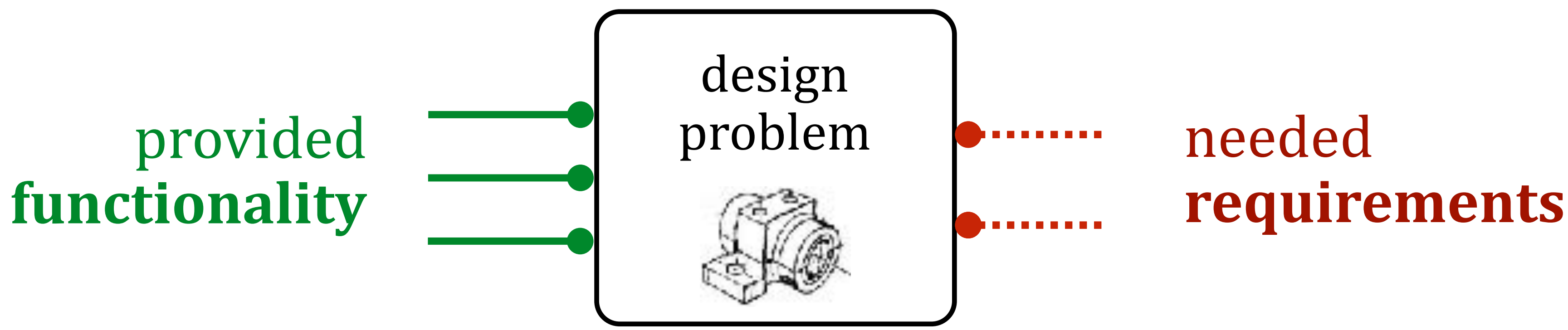
“costs”
“dependencies”

fortunes



debts

- A **design problem** is a **relation** between **provided functionality** and **needed requirements.**



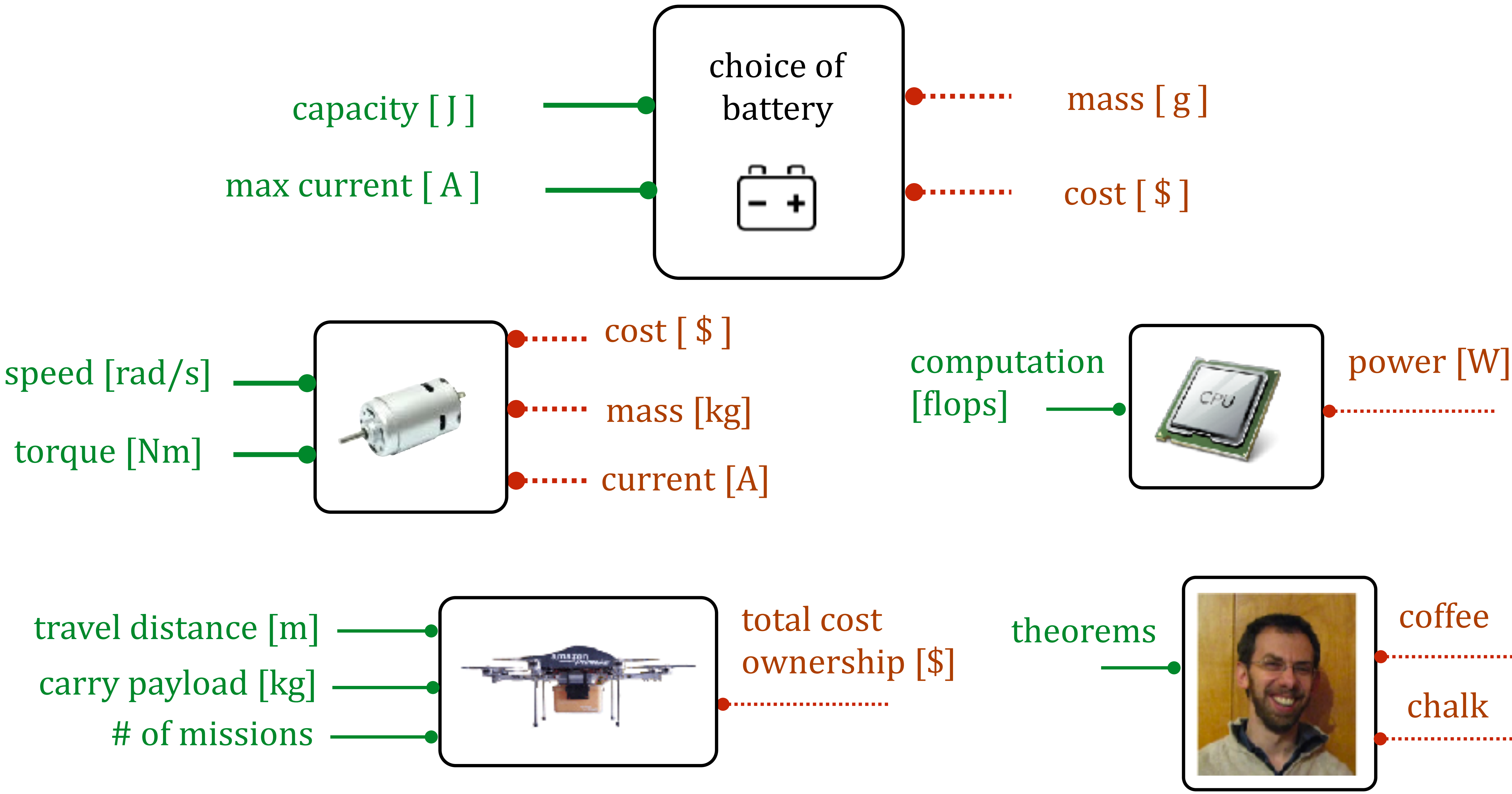
$\langle \mathcal{F}, \leq_{\mathcal{F}} \rangle$

any partially ordered set

$\langle \mathcal{R}, \leq_{\mathcal{R}} \rangle$

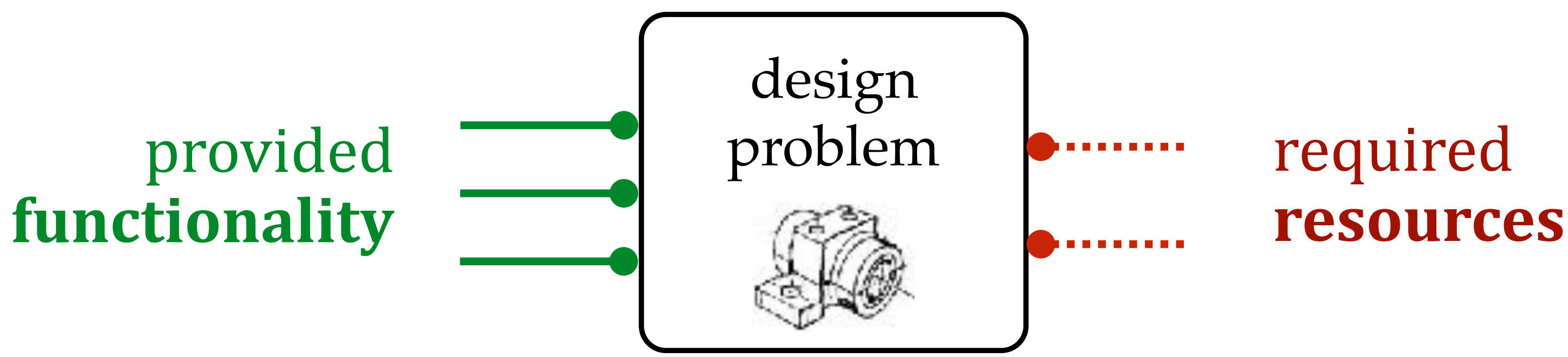
any partially ordered set

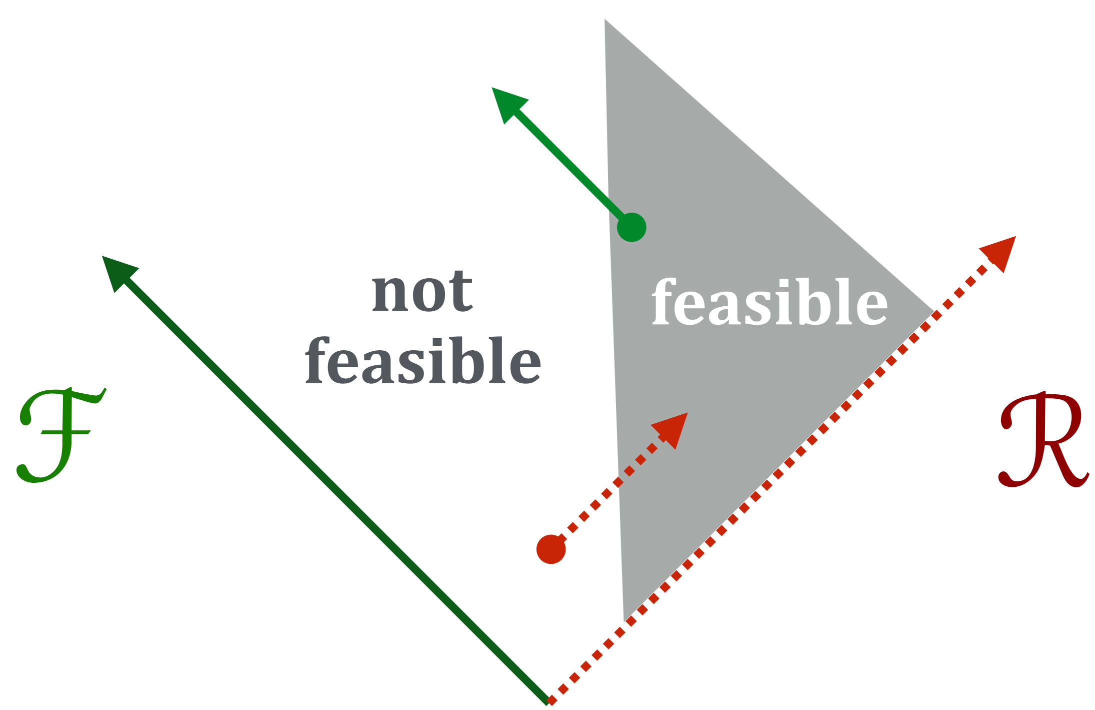
- A **design problem** is a **relation** between **provided functionality** and **needed requirements.**



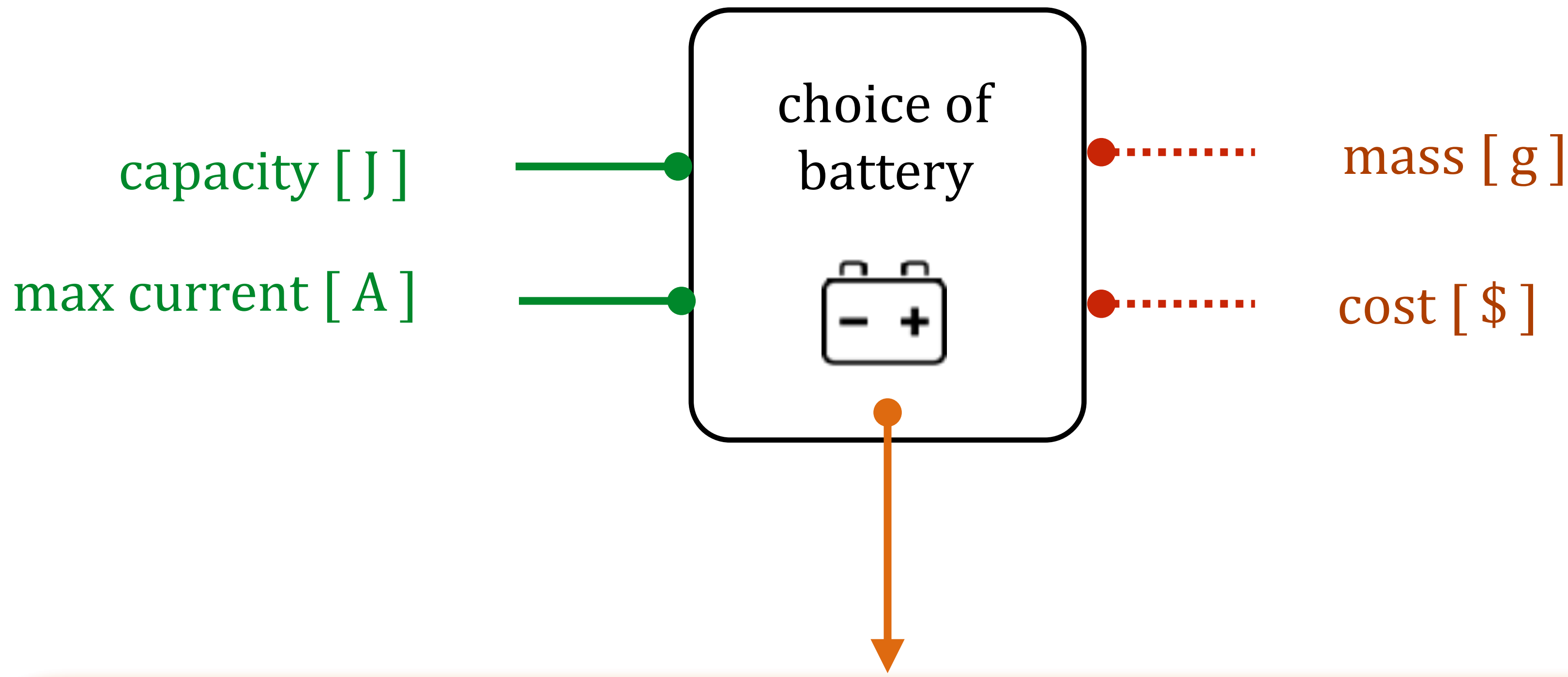
- A **design problem** is a "**monotone**" **relation** between a poset of **provided functionality** and a poset of **required resources.**

$$\mathrm{dp}\colon \mathcal{F}^{\mathrm{op}} \times \mathcal{R} \to_{\mathbf{Pos}} \mathrm{Bool}$$

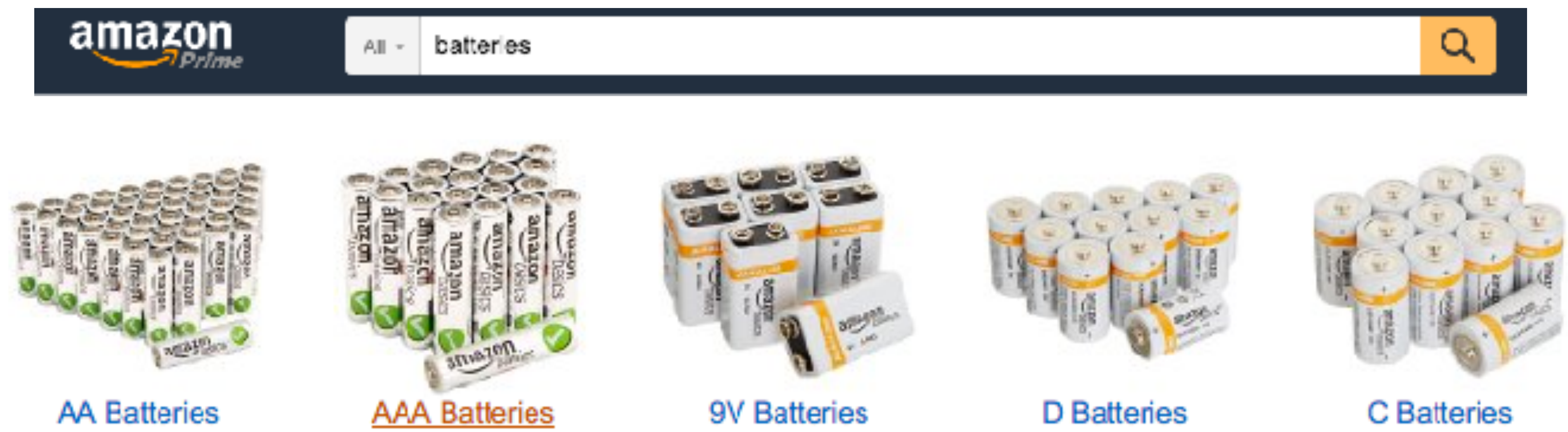




- A **design problem** is a **relation** between **provided functionality**, **required resources**, and **implementations**.



implementations



Is proof relevance relevant?

- **Engineering is "constructive":** for the purpose of design, **we need to know <u>how</u> something is done.**
- Morphisms are morally generated by **spans** of this type:

$$\begin{tikzcd}[column sep=small] & \mathcal{I} \text{ any set} \arrow[dl, "\alpha : \mathcal{I} \to \mathcal{F}"'] \arrow[dr, "\beta : \mathcal{I} \to \mathcal{R}"] & \\ \mathcal{F} \text{ a poset} & & \mathcal{R} \text{ a poset} \end{tikzcd} \longrightarrow$$

"which implementations are feasible?"

$$f : \mathcal{F}^{\mathrm{op}} \times \mathcal{R} \to_{\mathbf{Pos}} \mathrm{subsets}(\mathcal{I})$$
$$\langle f^{\mathrm{op}}, \mathrm{r} \rangle \mapsto \{ i \in \mathcal{I} : (f \leq_{\mathcal{F}} \alpha(i)) \wedge (\beta(i) \leq_{\mathcal{R}} r) \}$$

- But, **for the purpose of global computation, components only interact through the interfaces**. So we can look at Boolean pro-functors.

$$f : \mathcal{F}^{\mathrm{op}} \times \mathcal{R} \to_{\mathbf{Pos}} \mathrm{subsets}(\mathcal{I}) \xrightarrow[\mathbf{Pos}]{\text{nonempty?}} \mathbf{Bool}$$

In the **category DP**, also known as $\mathbf{Feas} = \mathbf{Prof}_{\mathbf{Bool}}$
Objects are posets, morphisms are Boolean profunctors

$$f : \mathcal{F}^{\mathrm{op}} \times \mathcal{R} \to_{\mathbf{Pos}} \mathbf{Bool}$$

also written: $f : \mathcal{F} \nrightarrow \mathcal{R}$

linear logic-like notation:

$$f : \mathcal{R} \multimap \mathcal{F}$$ ***(oops!)***

- For some algorithms, we will need more assumptions (e.g. objects are DCPOs)

Is proof relevance **relevant**?

- **Engineering is "constructive":** for the purpose of design, **we need to know <u>how</u> something is done.**
- Morphisms are morally generated by **spans** of this type:

$$\begin{tikzcd} & \mathcal{I} \text{ any set} \arrow[dl, "\alpha : \mathcal{I} \to \mathcal{F}"'] \arrow[dr, "\beta : \mathcal{I} \to \mathcal{R}"] & \\ \mathcal{F} \text{ a poset} & & \mathcal{R} \text{ a poset} \end{tikzcd}$$

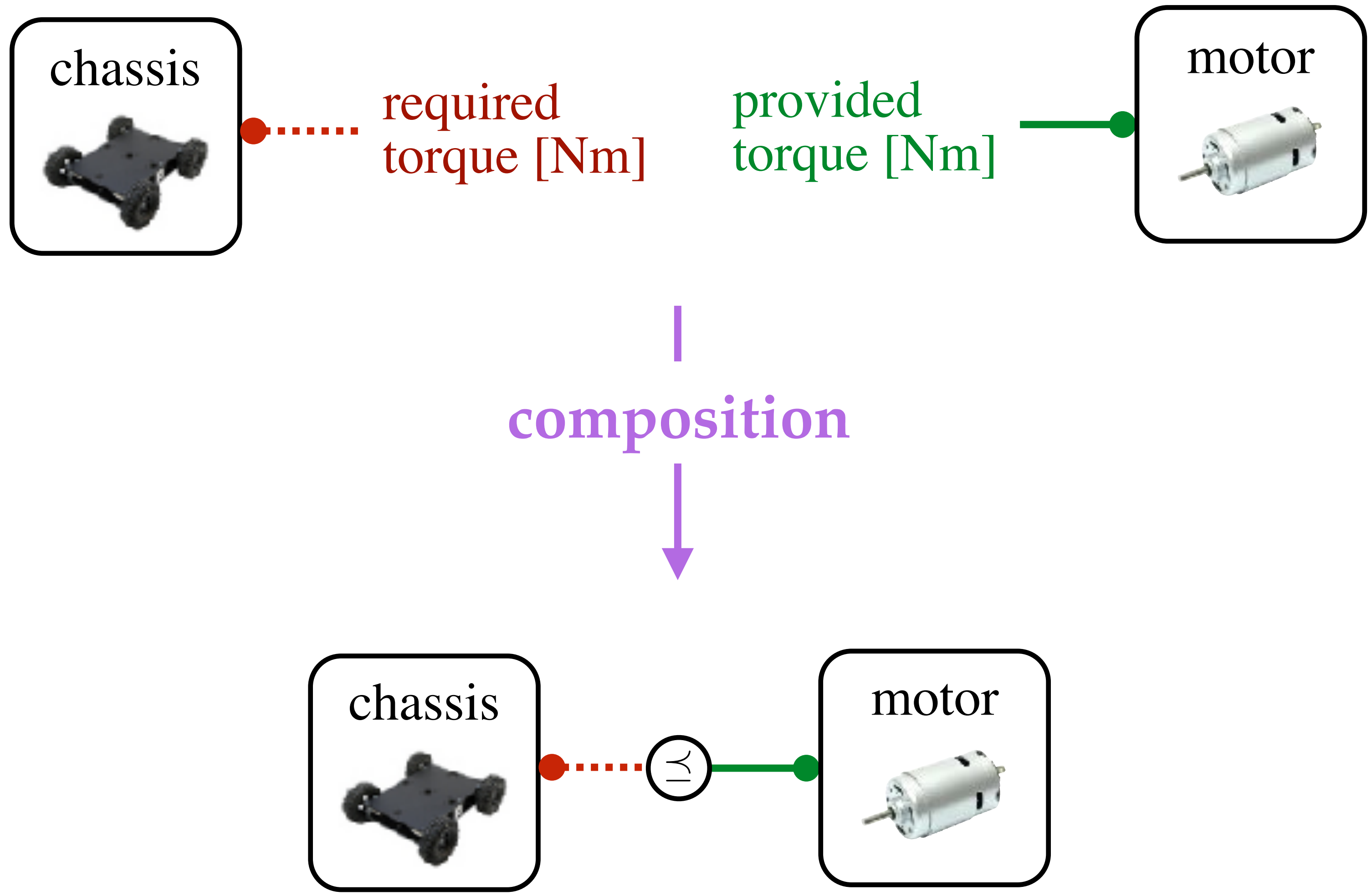
$\longrightarrow$

"which implementations are feasible?"

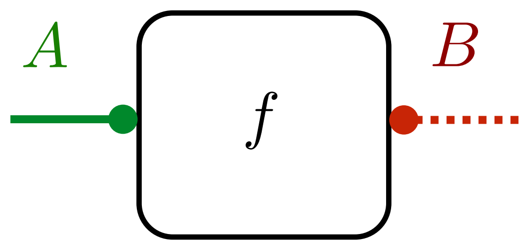
$$f : \mathcal{F}^{\text{op}} \times \mathcal{R} \to_{\mathbf{Pos}} \text{subsets}(\mathcal{I})$$

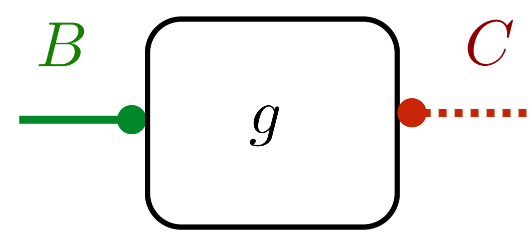
$$\langle f^{\text{op}}, \mathsf{r} \rangle \mapsto \{ i \in \mathcal{I} : (f \leq_{\mathcal{F}} \alpha(i)) \wedge (\beta(i) \leq_{\mathcal{R}} r) \}$$

- **If the rest of the talk is too slow** for you:
  - You can re-do all the **constructions with span**s.
  - You can think about the **Curry-Howard isomorphism**.
    (try resources = propositions)
  - You can think about the fact that all is entirely **symmetric**.
  - Nice Haskell you got there, but is it **reversible**?

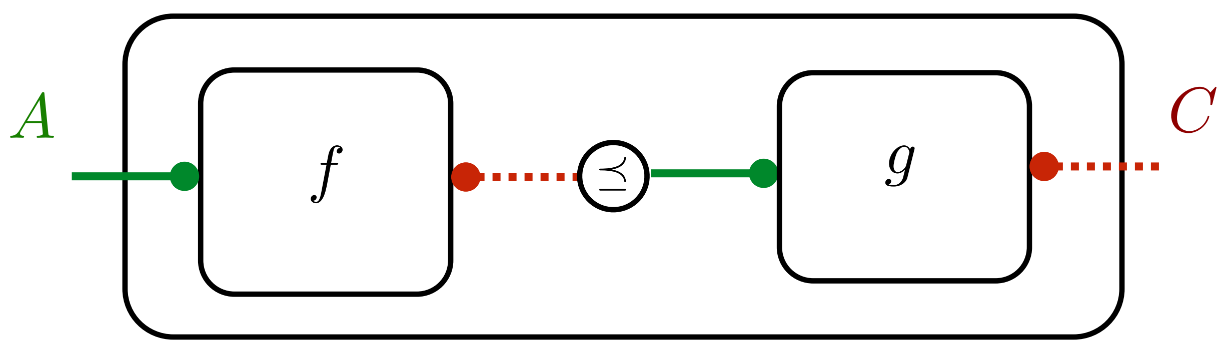


resources required by the first system $\preceq$ **functionality provided by the second system**





$$\frac{f : A \nrightarrow B \qquad\qquad g : B \nrightarrow C}{(f\,;g) : A \nrightarrow C}$$



- Profunctor composition:

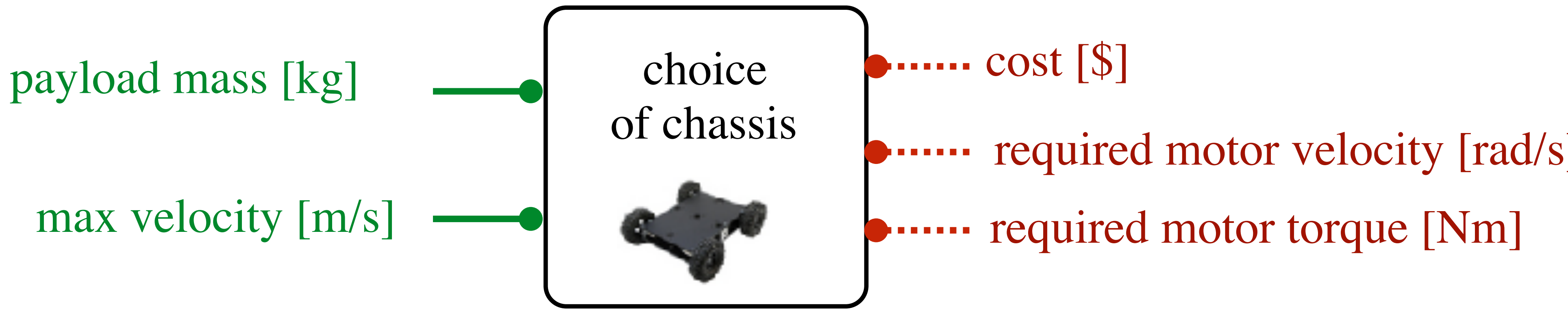
$$\frac{f : A^{\mathrm{op}} \times B \to_{\mathbf{Pos}} \mathbf{Bool} \qquad\qquad g : B^{\mathrm{op}} \times C \to_{\mathbf{Pos}} \mathbf{Bool}}{(f;\ g) : A^{\mathrm{op}} \times C \to_{\mathbf{Pos}} \mathbf{Bool}}$$

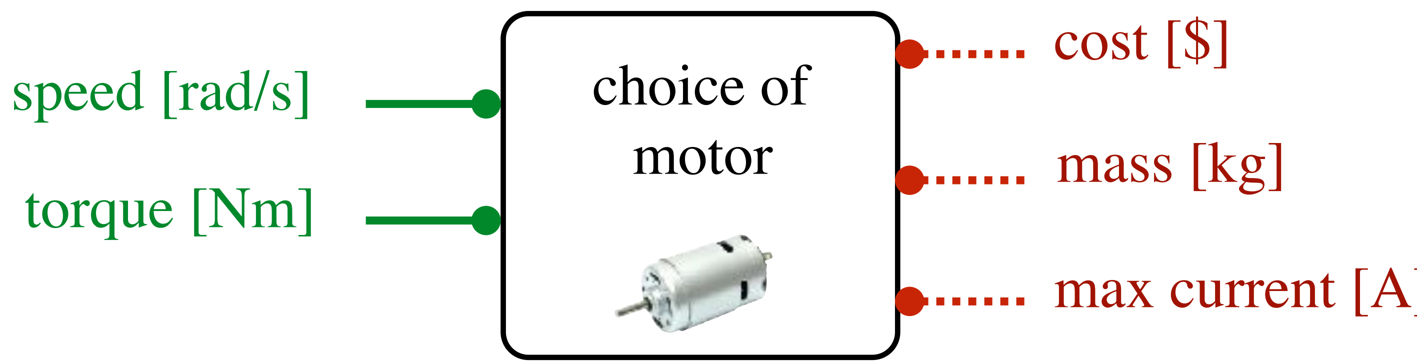
$$\langle a^{\mathrm{op}}, c\rangle \mapsto \bigvee_{b_1 \leq b_2} f(a^{\mathrm{op}}, b_1) \wedge g(b_2^{\mathrm{op}}, c)$$

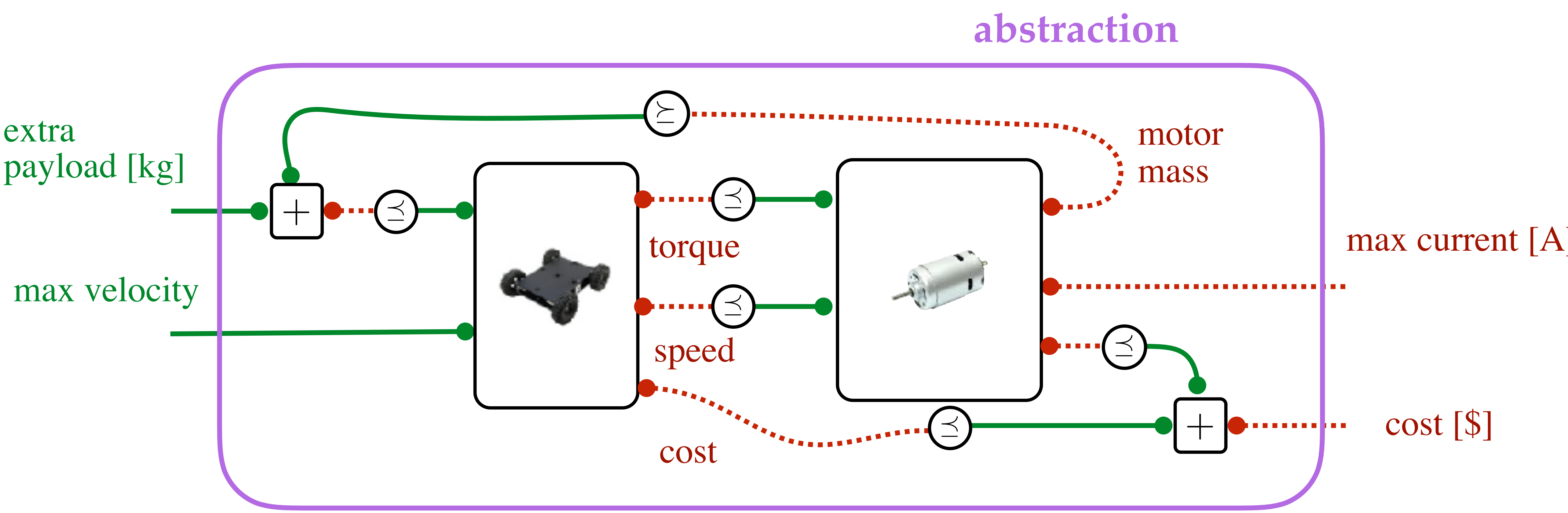
- What is the identity?

$$\mathrm{Id}_A : A^{\mathrm{op}} \times A \to_{\mathbf{Pos}} \mathbf{Bool}$$

$$\langle {a_1}^{\mathrm{op}}, a_2\rangle \ \mapsto \ (a_1 \leq_A a_2)$$

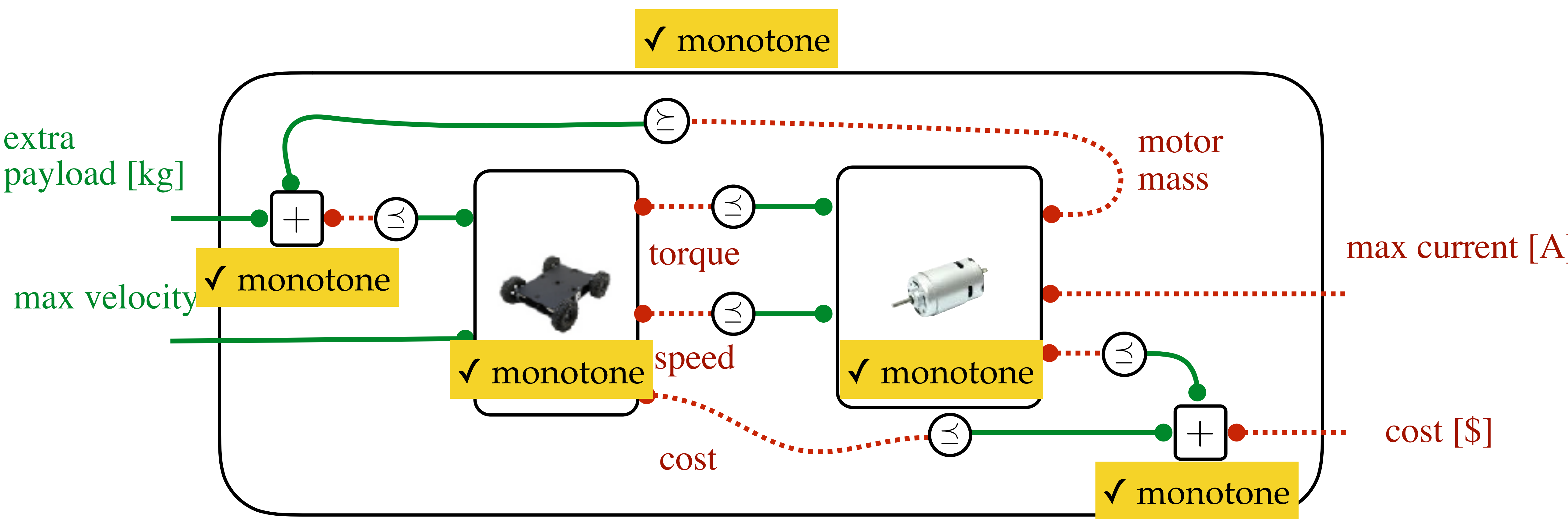

payload mass [kg]
max velocity [m/s]
choice of chassis
cost [$]
required motor velocity [rad/s]
required motor torque [Nm]


speed [rad/s]
torque [Nm]
choice of motor
cost [$]
mass [kg]
max current [A]

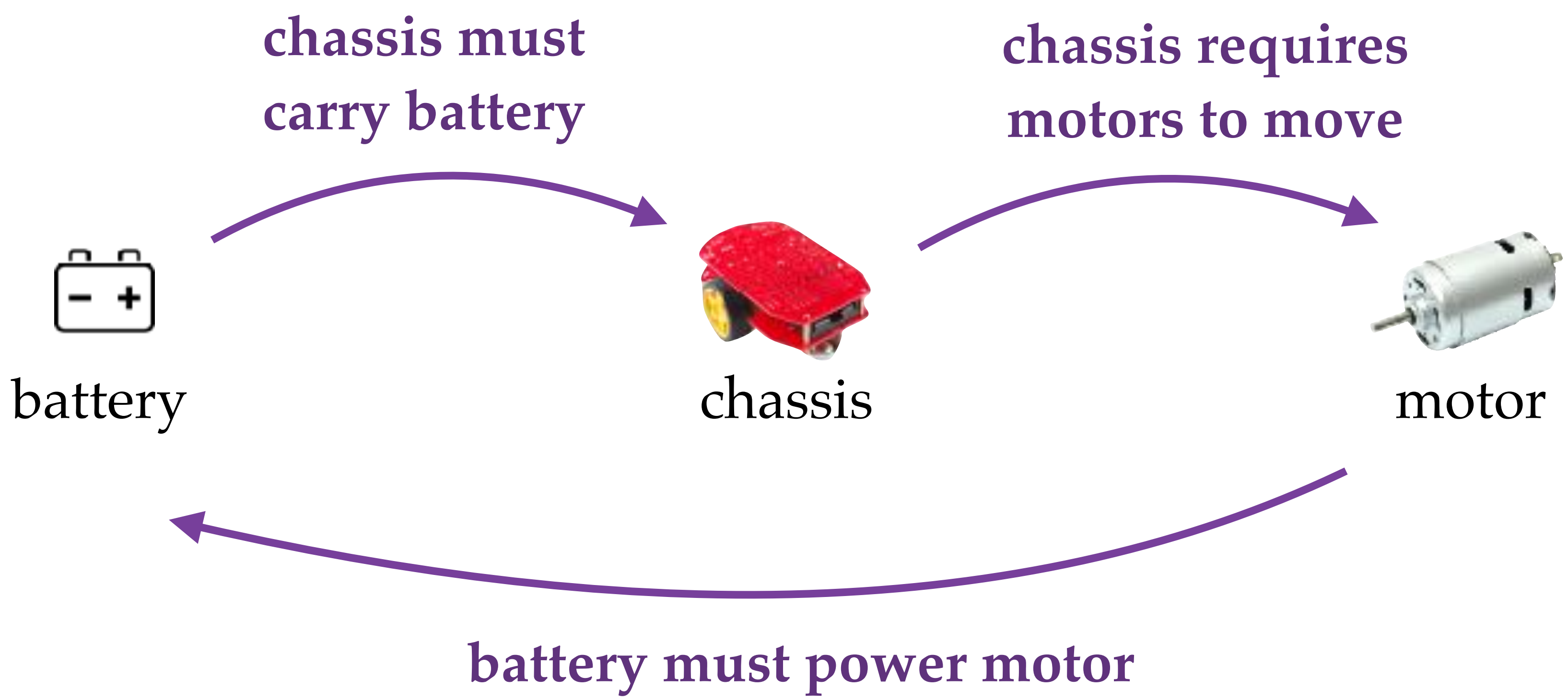

abstraction
extra payload [kg]
max velocity
motor mass
torque
speed
cost
max current [A]
cost [$]

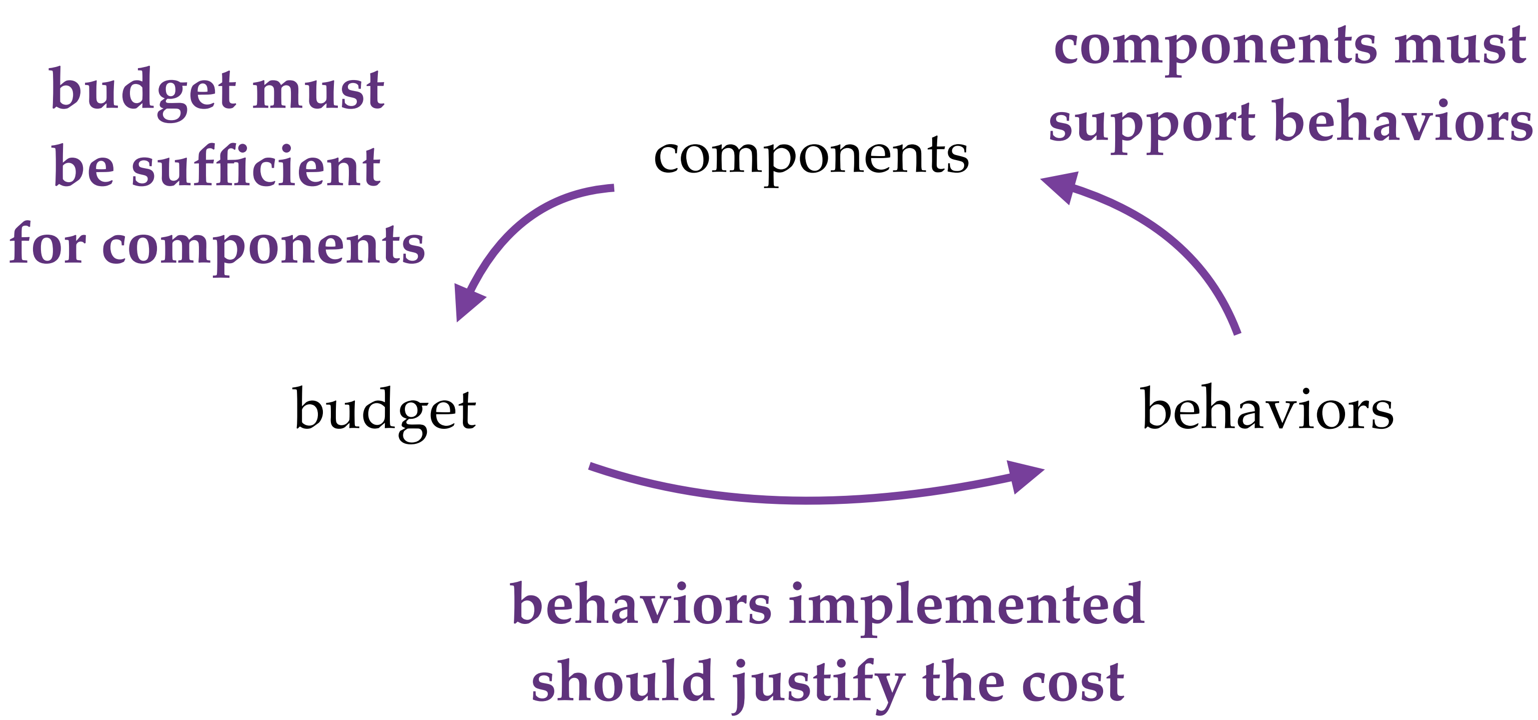
- Surprising result (at the time):

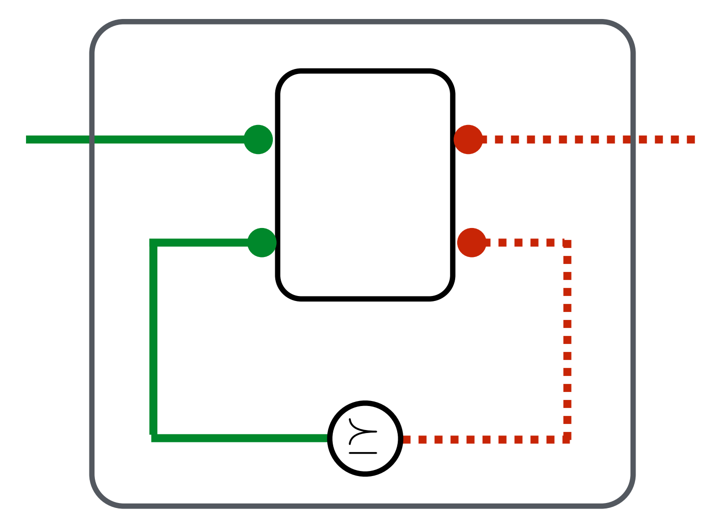
  The interconnection of any number of monotone design problems is monotone.



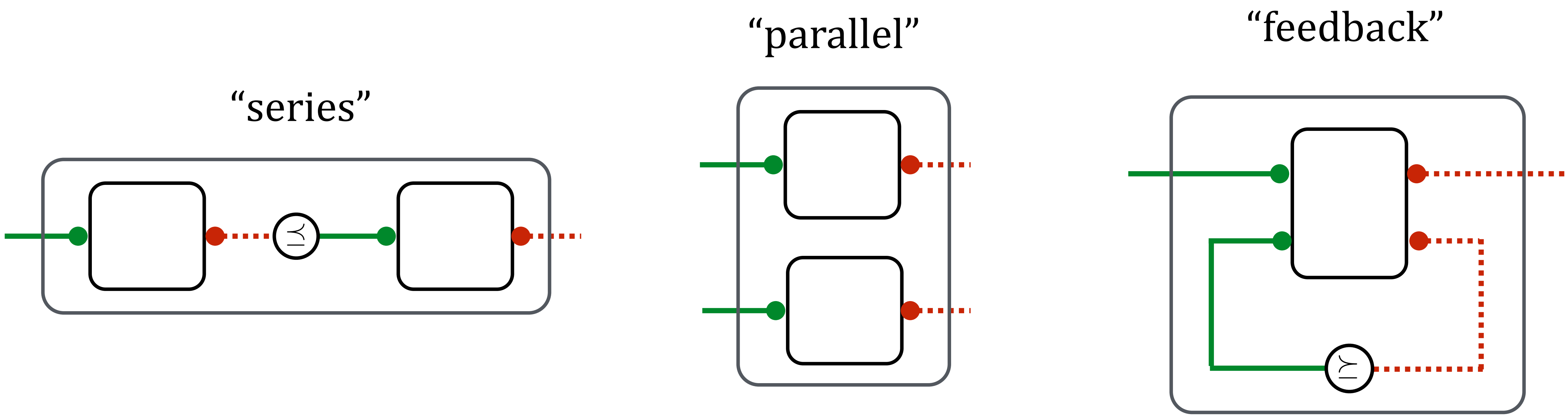
Feedback = irreducible complexity of design







- Recap: structure of **traced, symmetric, monoidal category**



- Also: locally posetal; actually **enriched in BoundedLat**

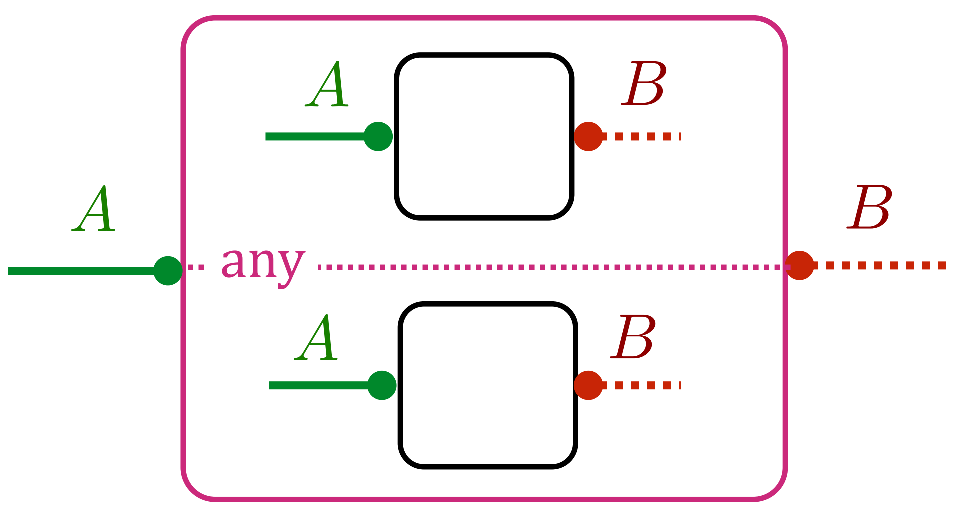
$$\top_{AB} \;\geq_{AB}\; f \cup_{AB} g \;\geq_{AB}\; f \cap_{AB} g \;\geq_{AB}\; \bot_{AB}$$

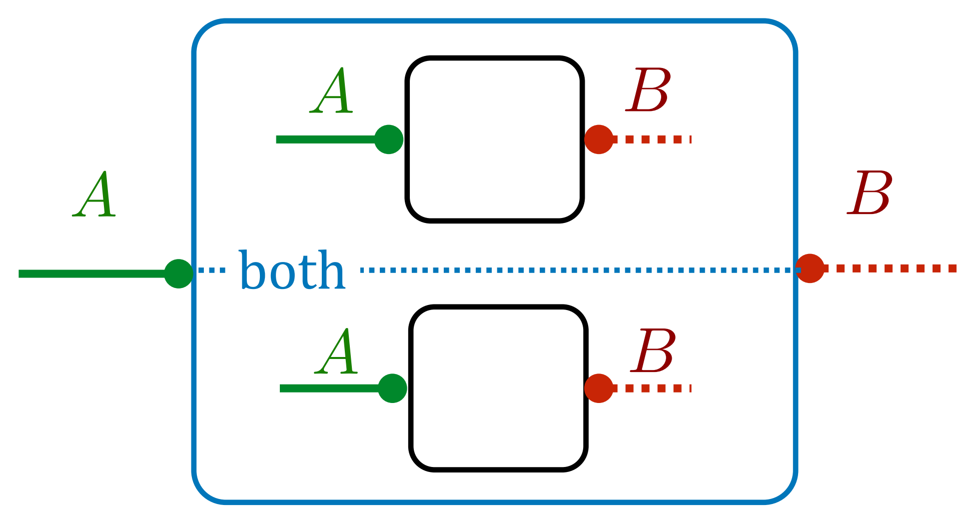
“everything is possible”

“choose between two options”

“convince two experts”

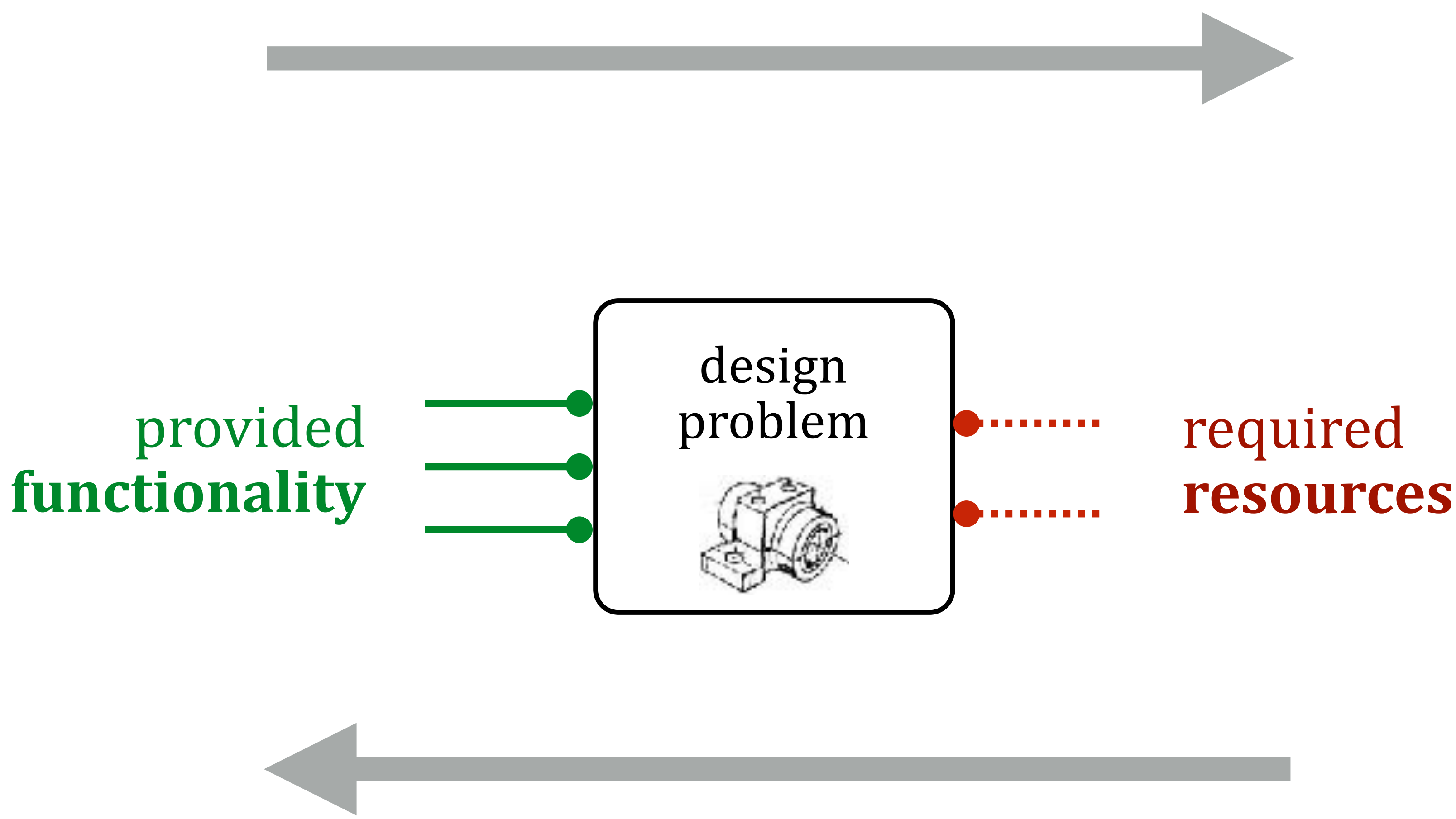
“nothing is possible”





- Objects are posets, hom-sets are lattices: a very “self-aware” category.

- **Given the functionality** to be provided, what are the **minimal resources** required?



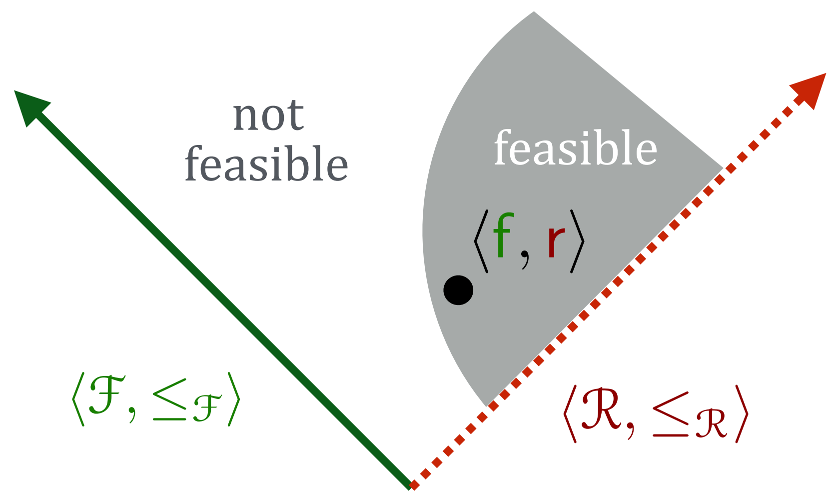
- **Given the resources** that are available, what is the **maximal functionality** that can be provided?

Assumptions for computation

- For each design problem, we have a representation in terms of antichains.

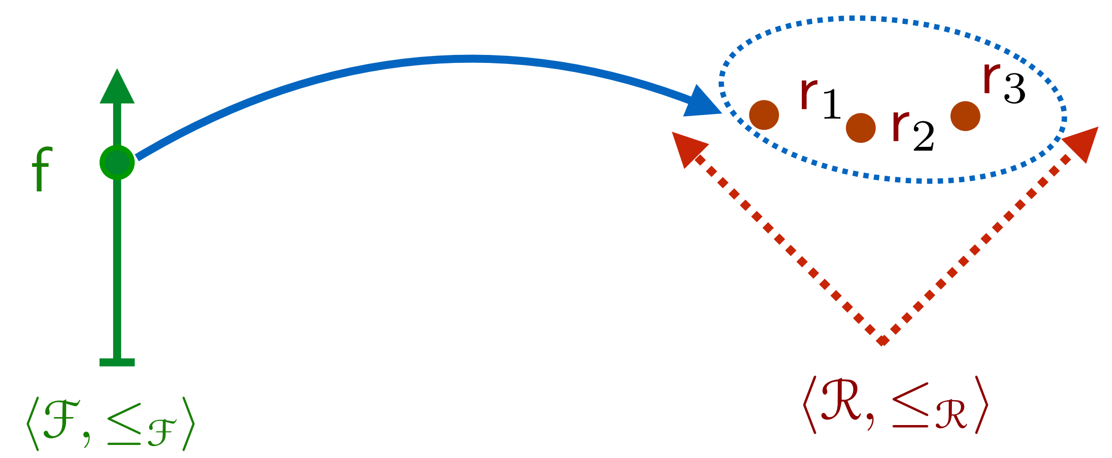
$$\mathrm{dp} : \mathcal{F}^{\mathrm{op}} \times \mathcal{R} \to_{\mathbf{Pos}} \mathbf{Bool}$$

"is this pair of (functionality, resources) feasible?"



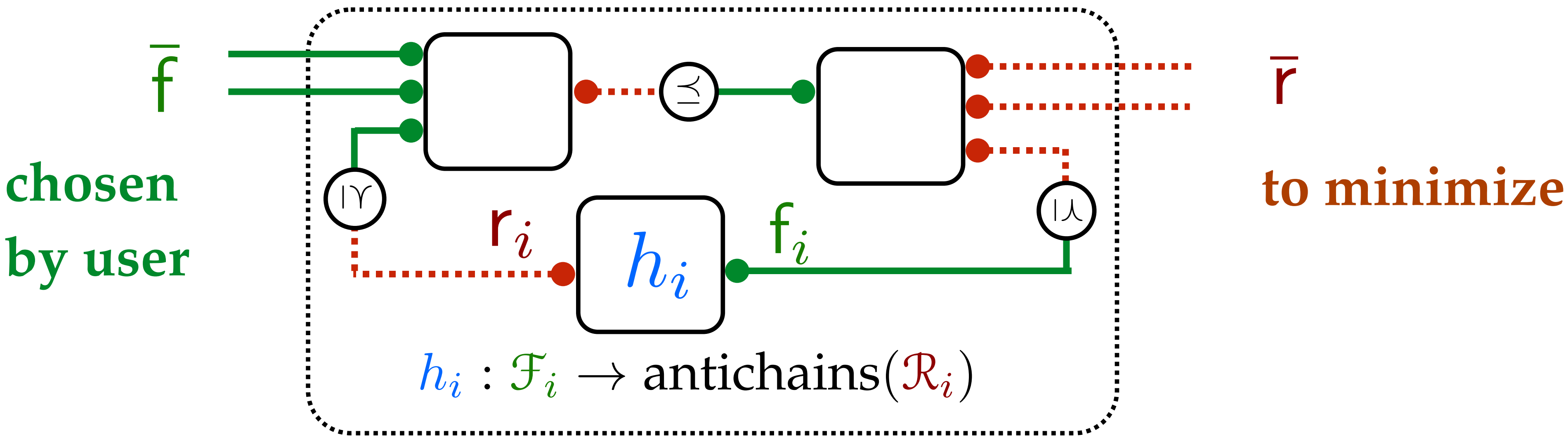
$$h : \mathcal{F} \to_{\mathbf{Pos}} \mathrm{antichains}(\mathcal{R})$$

"for a given functionality, what are the minimal resources necessary?"



- Assume: The posets are **pointed direct-complete partial orders.**
- Assume: The ***h*** maps are **Scott-continuous** ("continuous from the bottom").
- Under these assumptions, we can reduce the optimization problem to a **fixed-point formulation,** and use **Kleene's algorithm** to find the entire set of optimal solutions (or a certificate of infeasibility).

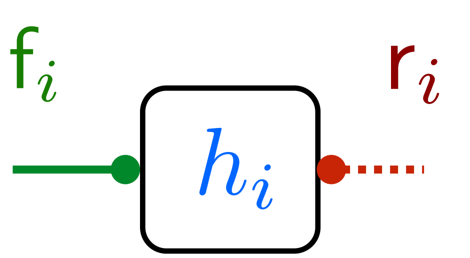
Semantics as an optimization problem



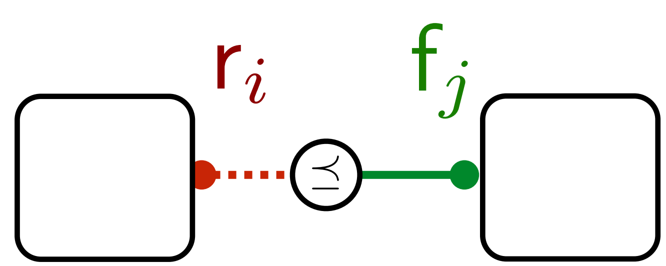
variables $r_i \in \langle \mathcal{R}_i, \preceq_{\mathcal{R}_i} \rangle$ $f_i \in \langle \mathcal{F}_i, \preceq_{\mathcal{F}_i} \rangle$

objective $\underset{\preceq}{\mathrm{Min}}\ \bar{r}$

constraints *for each node i:* *for each edge (i, j):*

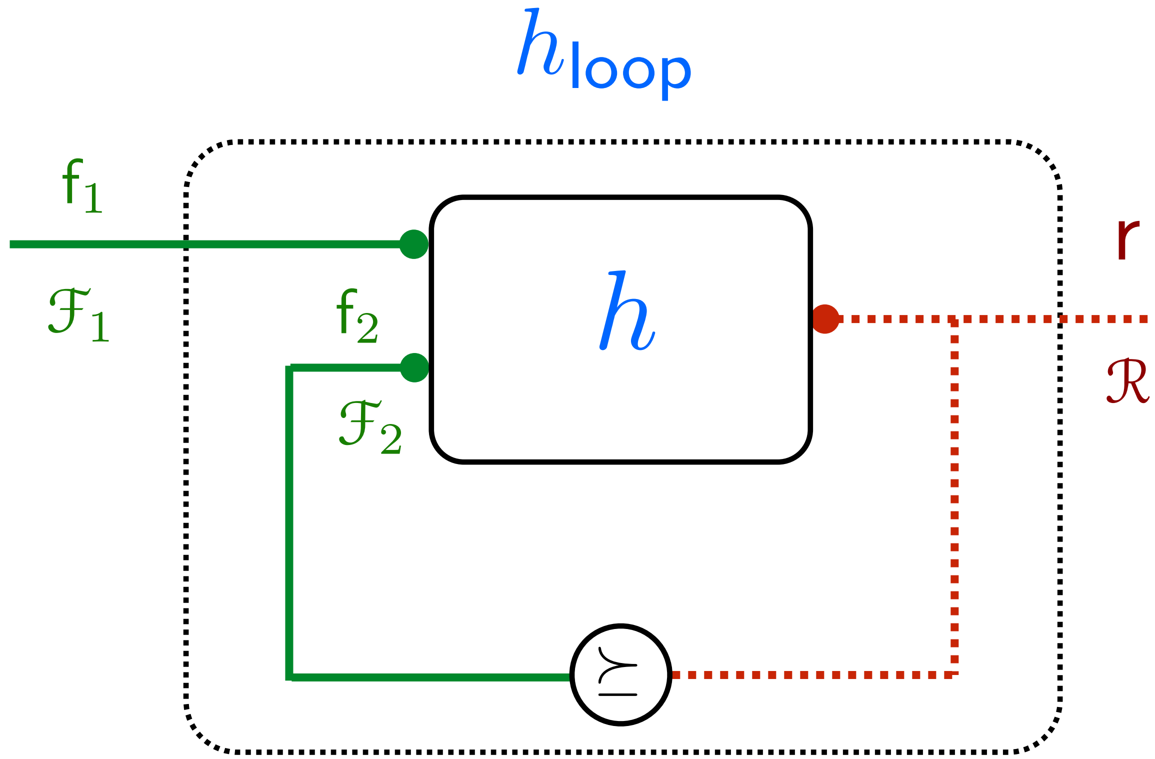


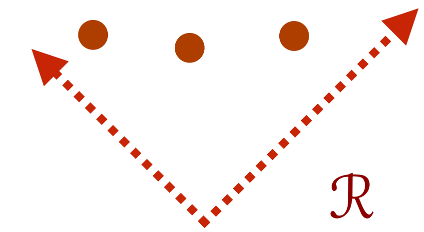
$r_i \in h_i(f_i)$



$f_j \succeq r_i$

- ! not convex
- ! not differentiable
- ! not continuous
- ! not even defined on continuous spaces

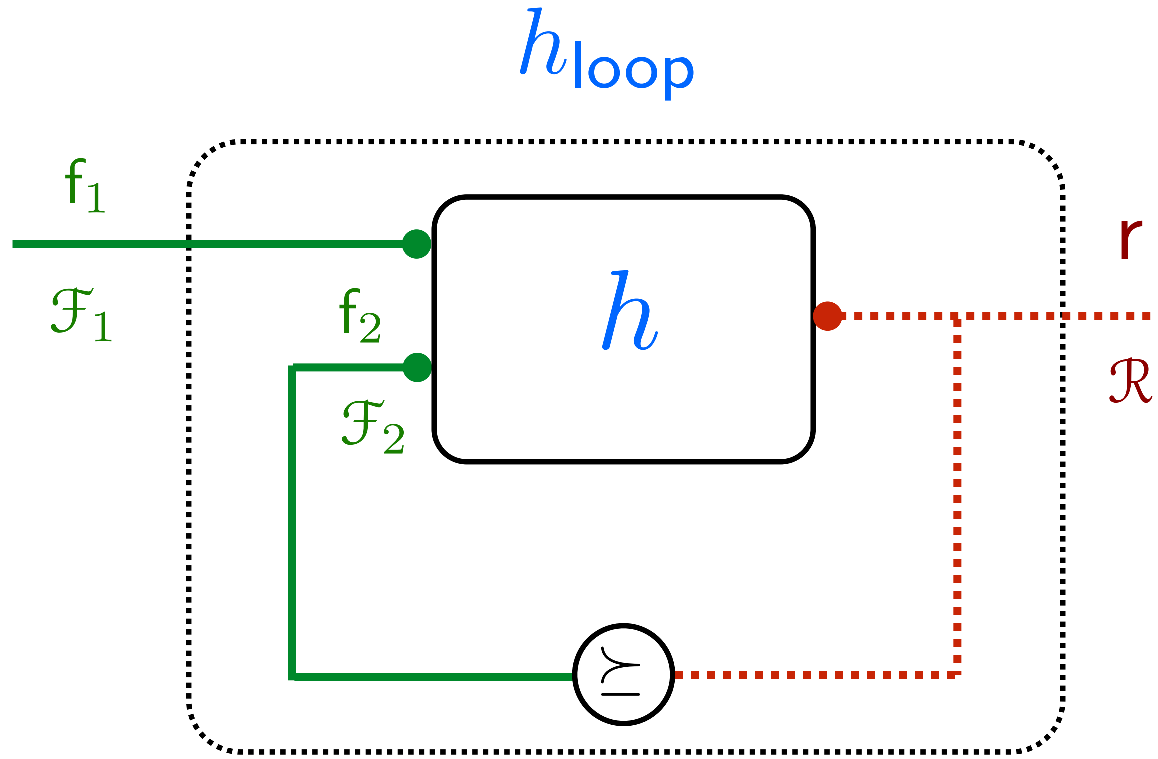


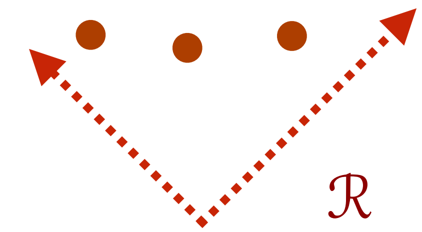


Theorem. The **set of minimal feasible resources** can be obtained as the least fixed point of a monotone function in the space of antichains.

$$\begin{aligned} h_{\mathsf{loop}} : \mathcal{F}_1 &\to \mathsf{antichains}(\mathcal{R}) \\ \mathsf{f}_1 &\mapsto \text{least-fixed-point}(\Phi_{\mathsf{f}_1}) \end{aligned}$$

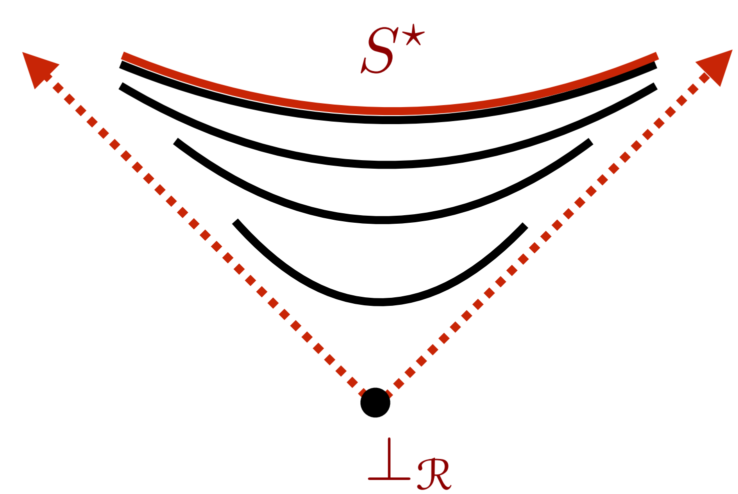
$$\begin{aligned} \Phi_{\mathsf{f}_1} : \mathsf{antichains}(\mathcal{R}) &\to \mathsf{antichains}(\mathcal{R}) \\ S &\mapsto \operatorname*{Min}_{\preceq_{\mathcal{R}}} \bigcup_{\mathsf{r} \in S} h(\mathsf{f}_1, \mathsf{r}) \cap \uparrow \mathsf{r} \end{aligned}$$





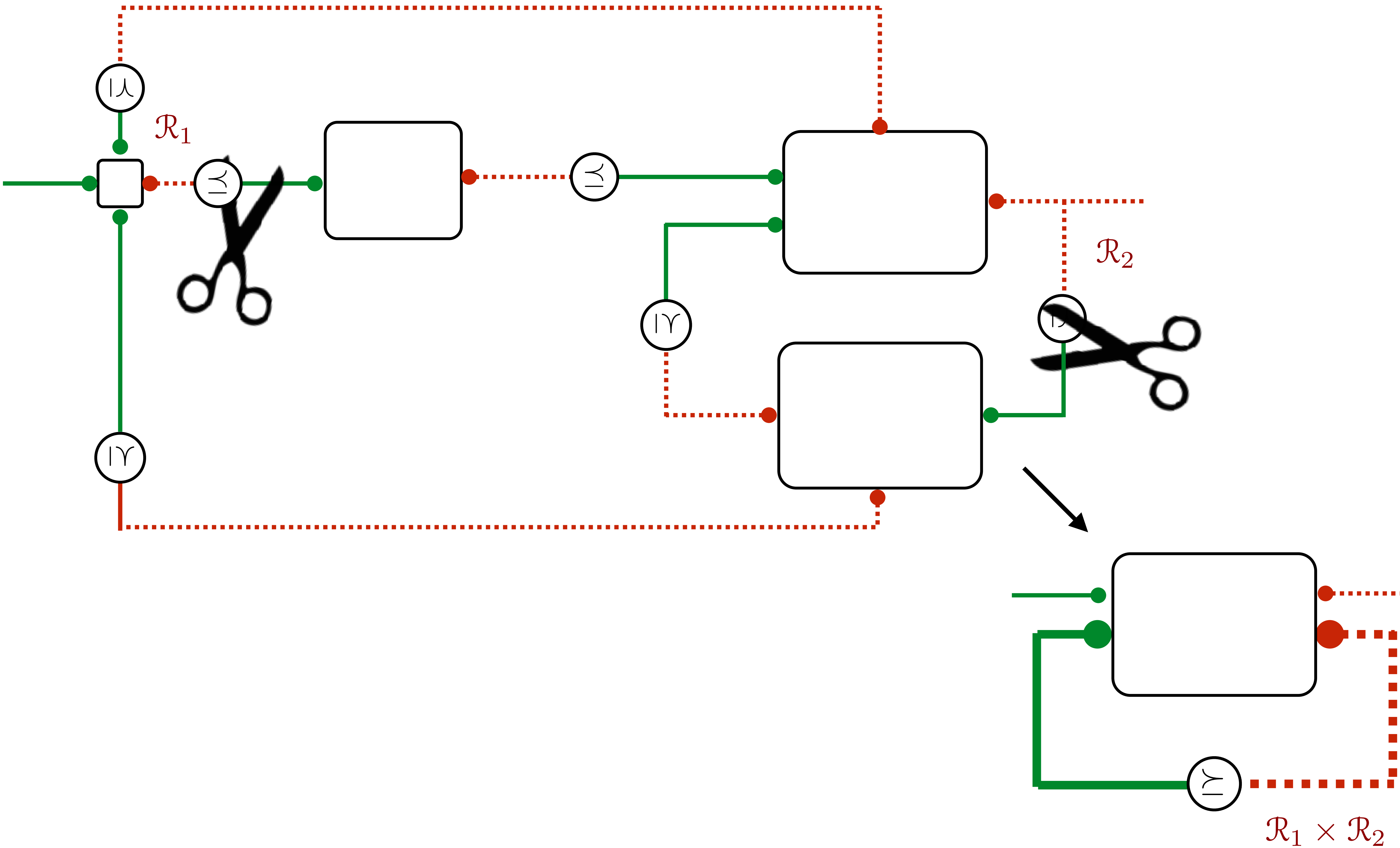
Corollary. The set of minimal solutions can be found using Kleene's algorithm.

$$S \subset \text{antichains}(\mathcal{R})$$
$$S_0 = \{\bot_{\mathcal{R}}\}$$
$$S_{k+1} = \Phi_{\mathsf{f}_1}(S_k)$$

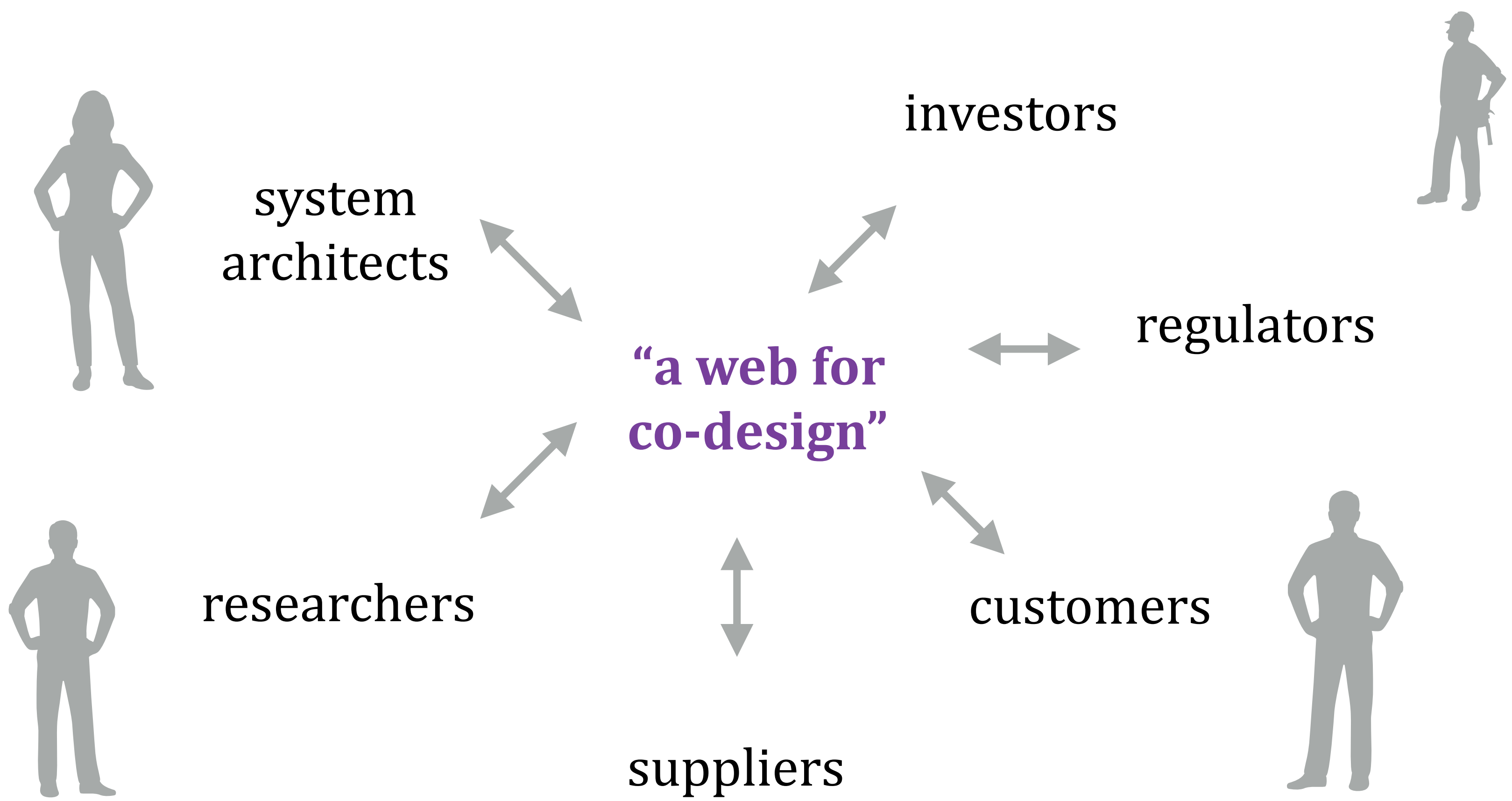


If the iteration diverges, it is a certificate of infeasibility.

- The complexity of solving the problem depends on the "thickness" of the "**minimal feedback arc set**".
  *(**Not combinatorial** in the size of the implementations!)*



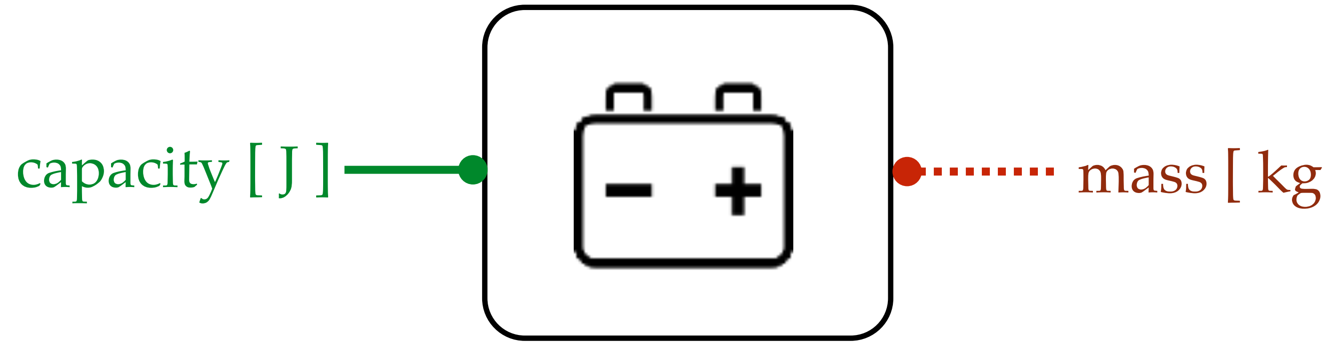
- A mathematical theory of co-design:
  - “co” for **compositional**
  - “co” for **computational**
  - “co” for **collaborative**
  - “co” for **continuous**



- What are the **languages and tools?**

Formal language

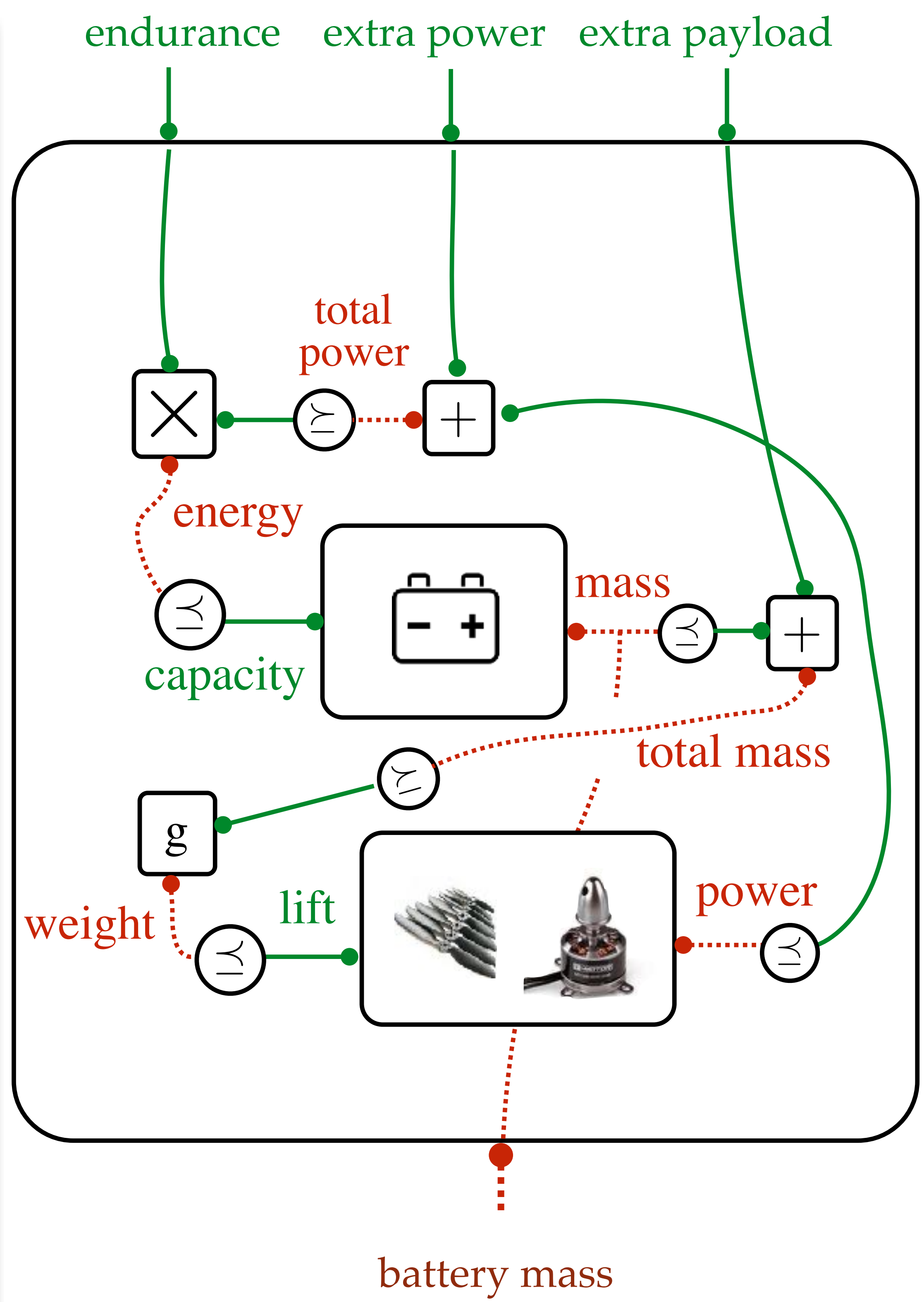
- I developed a **formal language for co-design**
    - inspired by Disciplined Convex Programming [Grant & Boyd]
- The user can only express monotone constraints
- Manual, IDE demo available at <u>http://demo.co-design.science</u>



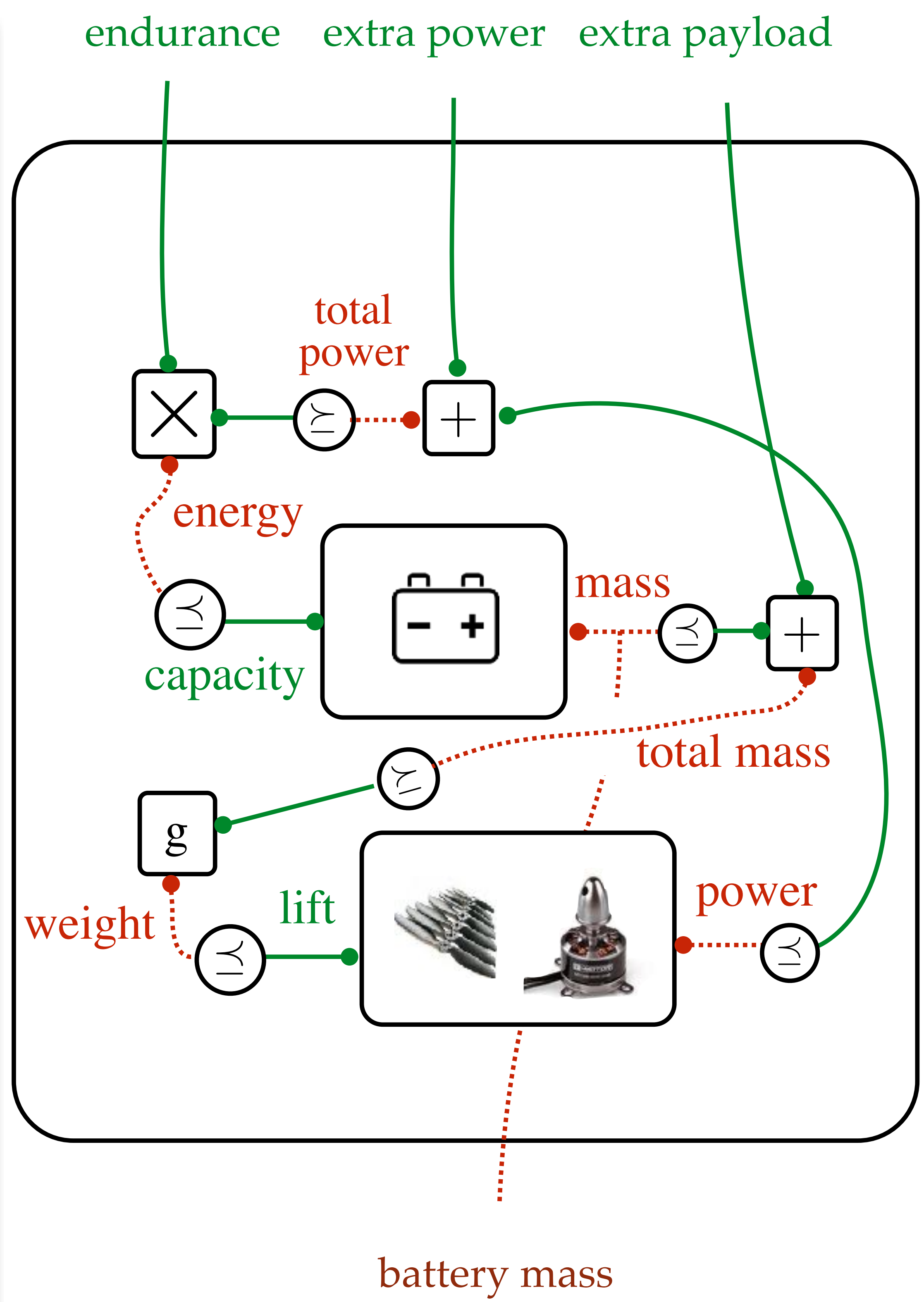

```
mcdp {
    provides capacity [J]
    requires mass [kg]

    specific_energy_Li_Ion = 500 Wh / kg

    mass >= capacity / specific_energy_Li_Ion
}
```


```
mcdp {
    provides lift [N]
    requires power [W]
    c = 10.0 W/N^2
    power >= c * lift^2
}
```

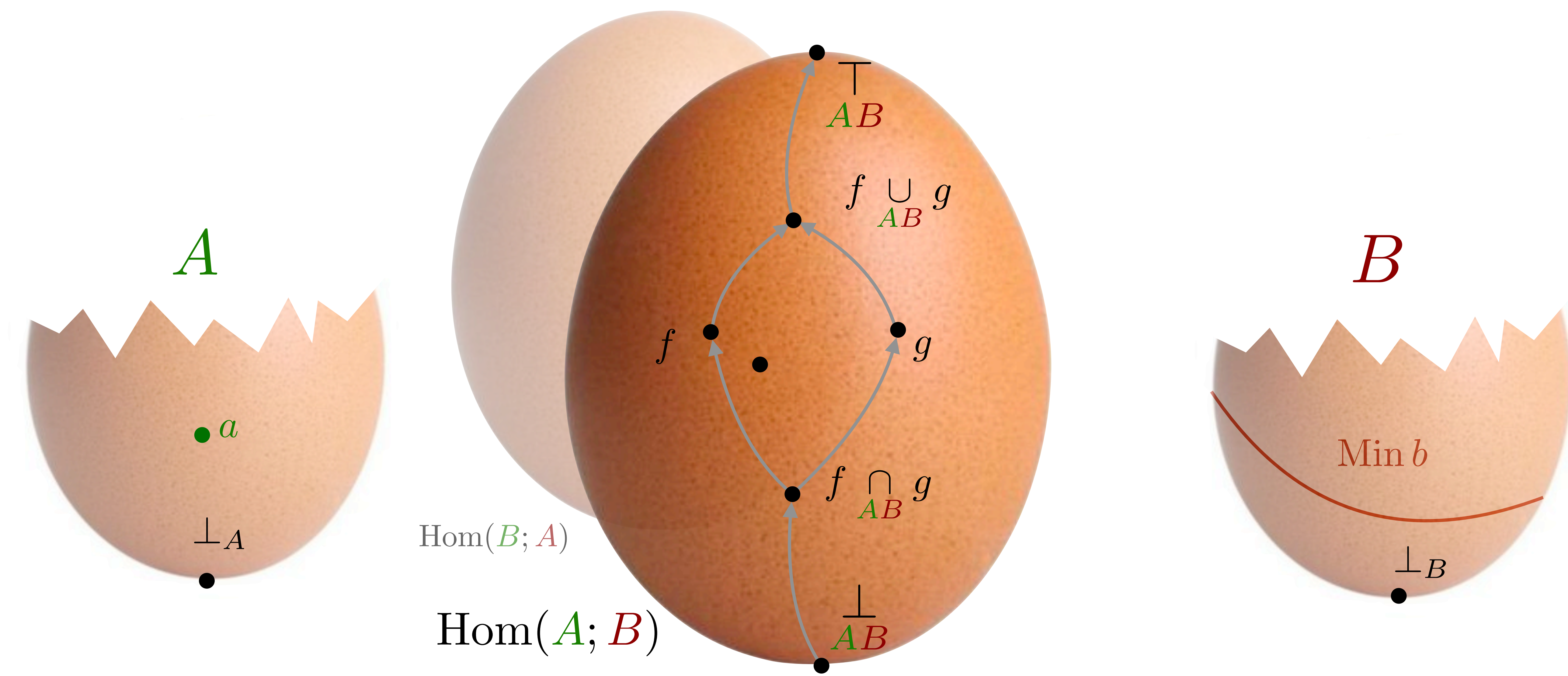
```
mcdp {
    # We need to fly for this duration
    provides endurance [s]
    # While carrying this extra payload
    provides extra_payload [kg]
    # And providing this extra power
    provides extra_power [W]

    # Sub-design problem: choose the battery
    sub battery = mcdp {
        # A battery provides capacity
        provides capacity [J]
        # and requires some mass to be transported
        requires mass [kg]
        # requires cost [$]

        specific_energy_Li_Ion = 500 Wh / kg

        mass >= capacity / specific_energy_Li_Ion
    }

    # Sub-design problem: actuation
    sub actuation = mcdp {
        # actuators need to provide this lift
        provides lift [N]
        # and will require power
        requires power [W]
        # simple model: quadratic
        c = 10.0 W/N^2
        power >= lift * lift * c
    }
    # Co-design constraint: battery must be large enough
    power = actuation.power + extra_power
    energy = power * endurance
    battery.capacity >= energy

    # Co-design constraint: actuators must be powerful enough
    gravity = 9.81 m/s^2
    weight = (battery.mass + extra_payload) * gravity
    actuation.lift >= weight

    # suppose we want to optimize for size of the battery
    requires mass for battery
}
```



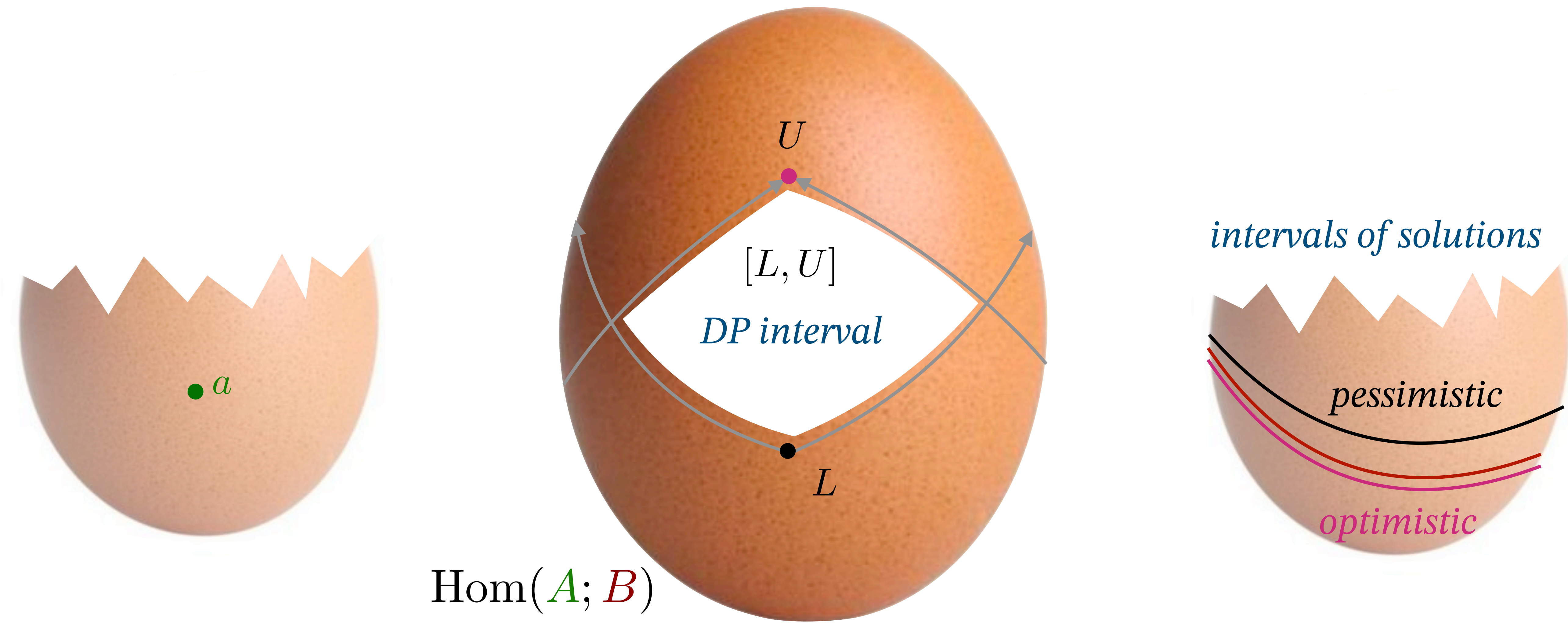
```
mcdp {
    # We need to fly for this duration
    provides endurance [s]
    # While carrying this extra payload
    provides extra_payload [kg]
    # And providing this extra power
    provides extra_power [W]

    # Sub-design problem: choose the battery
    sub battery = mcdp {
        # A battery provides capacity
        provides capacity [J]
        # and requires some mass to be transported
        requires mass [kg]
        # requires cost [$]

        specific_energy_Li_Ion = 500 Wh / kg

        mass >= capacity / specific_energy_Li_Ion
    }

    # Sub-design problem: actuation
    sub actuation = mcdp {
        # actuators need to provide this lift
        provides lift [N]
        # and will require power
        requires power [W]
        # simple model: quadratic
        c = 10.0 W/N^2
        power >= lift * lift * c
    }
    # Co-design constraint: battery must be large enough
    power = actuation.power + extra_power
    energy = power * endurance
    battery.capacity >= energy

    # Co-design constraint: actuators must be powerful enough
    gravity = 9.81 m/s^2
    weight = (battery.mass + extra_payload) * gravity
    actuation.lift >= weight

    # suppose we want to optimize for size of the battery
    requires mass for battery
}
```



Uncertainty in co-design



Uncertainty in co-design

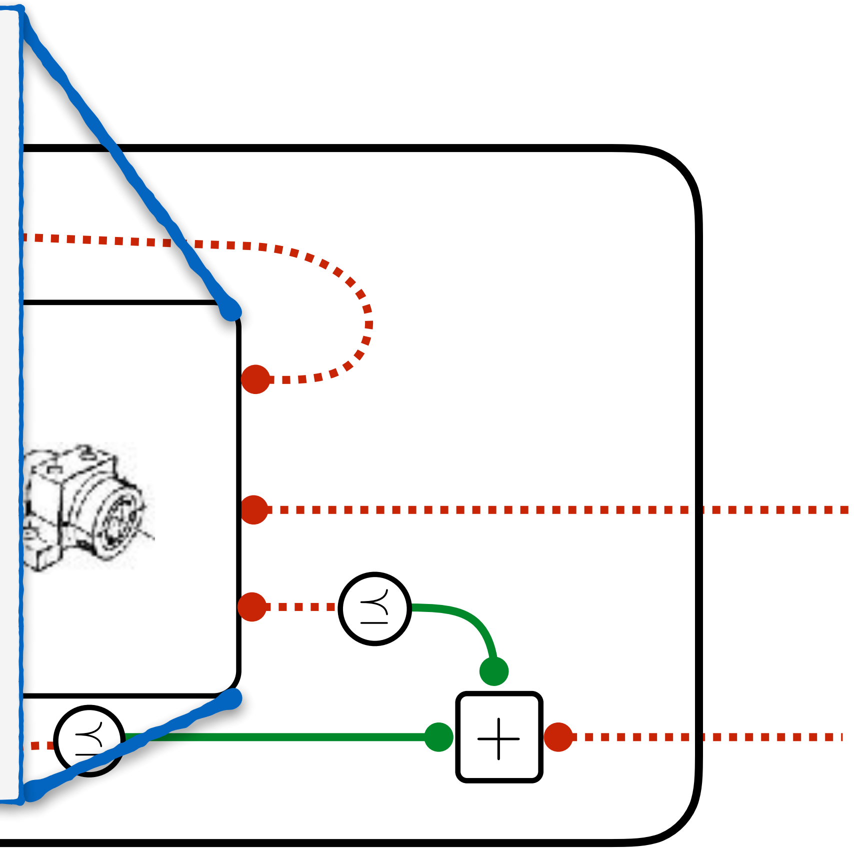


- A category of **Uncertain DPs:**
  - The objects are posets
  - The morphisms are ordered pairs of morphisms of DP.
- Everything generalizes easily.

Uncertainty in co-design problems

battery_uncertain.mcdp

```
mcdp {
    provides capacity [kWh]
    requires mass [g]
    requires cost [$]
    energy_density = between 140 kWh/kg and 150 kWh/kg
    specific_cost = 200 $/kWh
    required mass · energy_density ≥ provided capacity
    required cost ≥ provided capacity · specific_cost
}
```

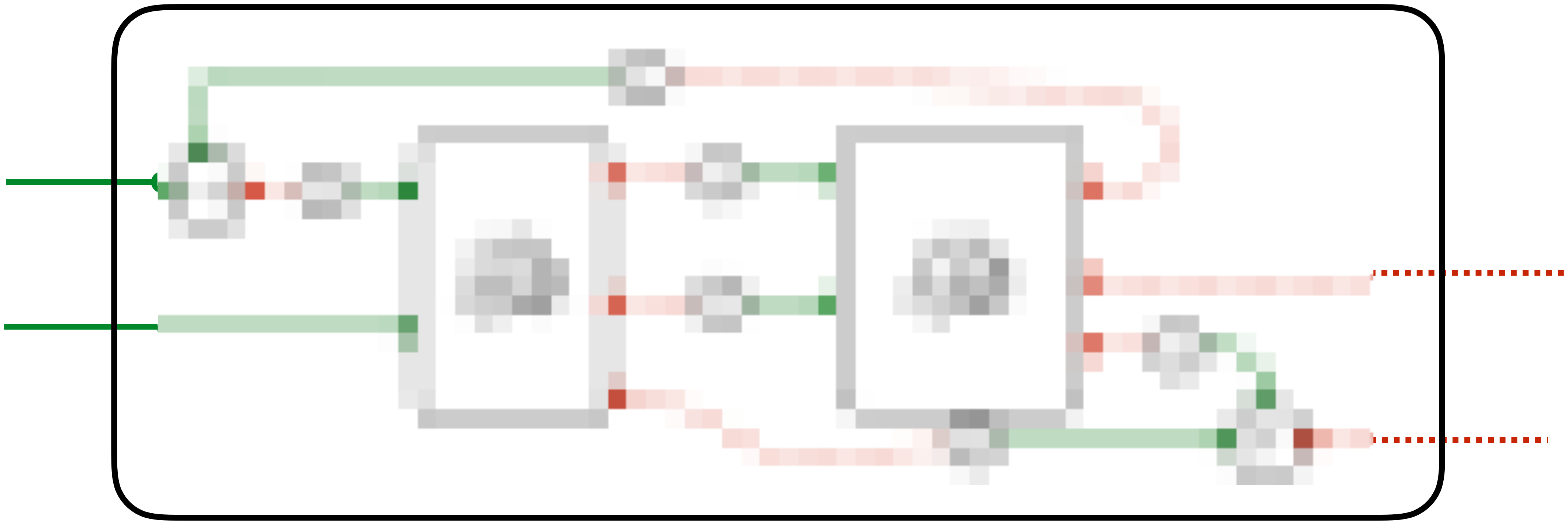


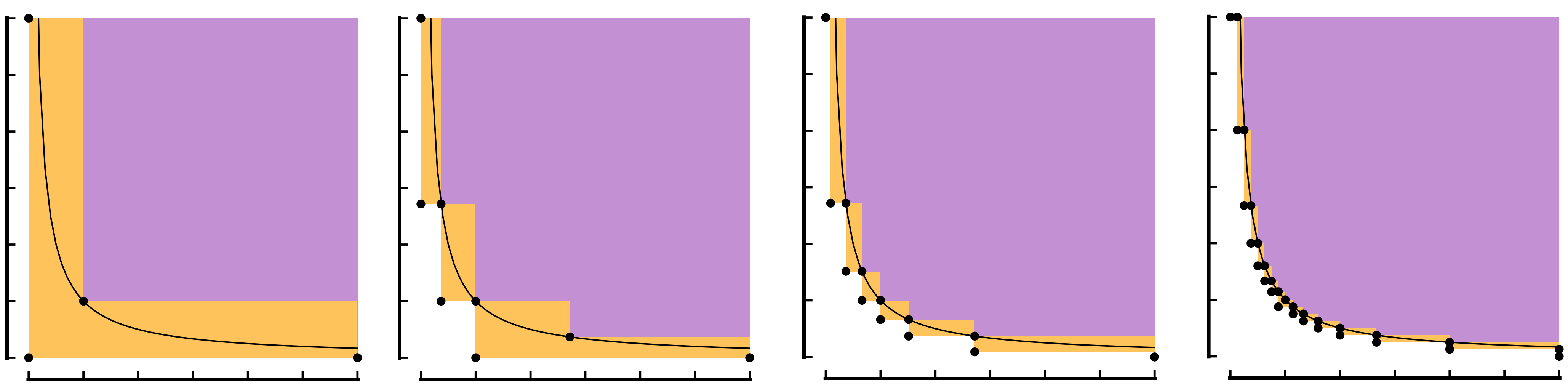
no uncertainty: "To obtain an endurance of **15 min**, the minimal cost is **$230**"

low uncertainty: "To obtain an endurance of **15 min**, the minimal cost is **between $220 and $240**"

high uncertainty: "To obtain an endurance of **15 min**, the minimal cost is **$220 in the best case, and in the worst case the problem is not feasible**"

Uncertainty for relaxation





Next steps for co-design theory

- The simple profunctor formalization does not capture all aspects of interest to the field of engineering design.

- **Seven more sketches** of more refined resource theories:
  1. **Monoidal and tropical** resources.
  2. **Homogeneous** resources
  3. **Higher-order** design theory.
  4. **Linearity** of resources.
  5. **Temporality** of resources.
  6. **Spatiality** of resources.
  7. **Negative co-design**.

- Note: **this is work in progress**, and open for collaboration. Let me know if this picks your fancy.

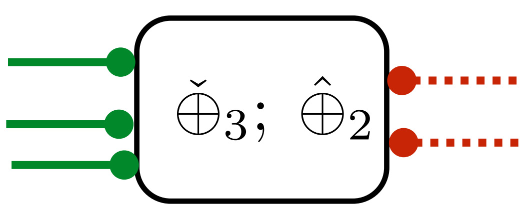
1. Monoidal and tropical resources

- For many resources posets, there is a “zero resource”, **resources can be added** together, etc.
- Assume the posets to have a **commutative monoid structure** compatible with the order.

$$(\oplus, 0) \qquad\qquad \frac{a \le b}{a \oplus c \le b \oplus c}$$

- You can generate for free a lot of interesting DPs.

$$a \oplus b \oplus c \le a \oplus b$$



$$\frac{\dfrac{f : A \nrightarrow B}{f^n : A^n \nrightarrow B^n}}{\tilde{f}^n : A \nrightarrow B}$$

- Careful, for some resources (e.g. time) you have both + and max:

$$a \oplus b = \max(a, b)$$

$$a \otimes b = a + b$$

“tropical semiring”

2. Homogeneous / heterogeneous resources

- Sometimes you want to further characterize the quality of resources:
  - "To make a car I need 4 identical wheels"
  - "We cannot go to the party if we are dressed the same"

- Say that A is a type with **equality** and **apartedness.**

$x =_A y$ *type of proofs that x and y are equal at A*

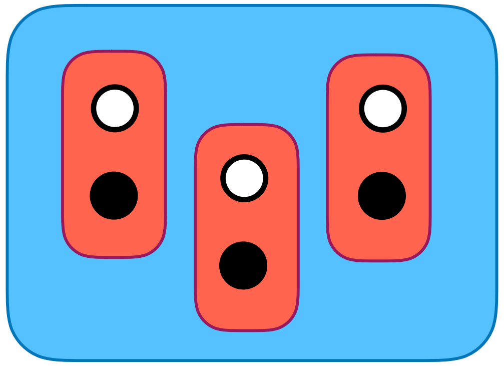
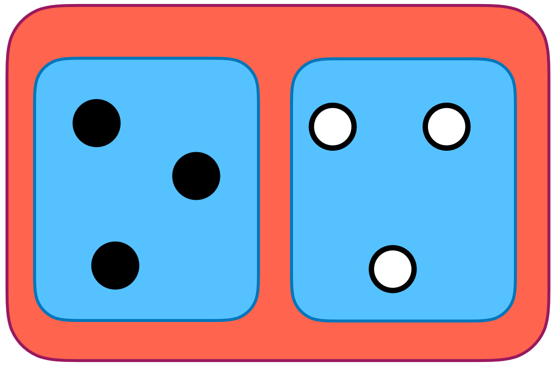
$x \neq_A y$ *type of proofs that x and y are different at A*

these proofs are also resources!

$$A_{\neq 2} \doteq (x : A) \times (y : A) \times (p : x \neq_A y)$$ *two certifiably different As*

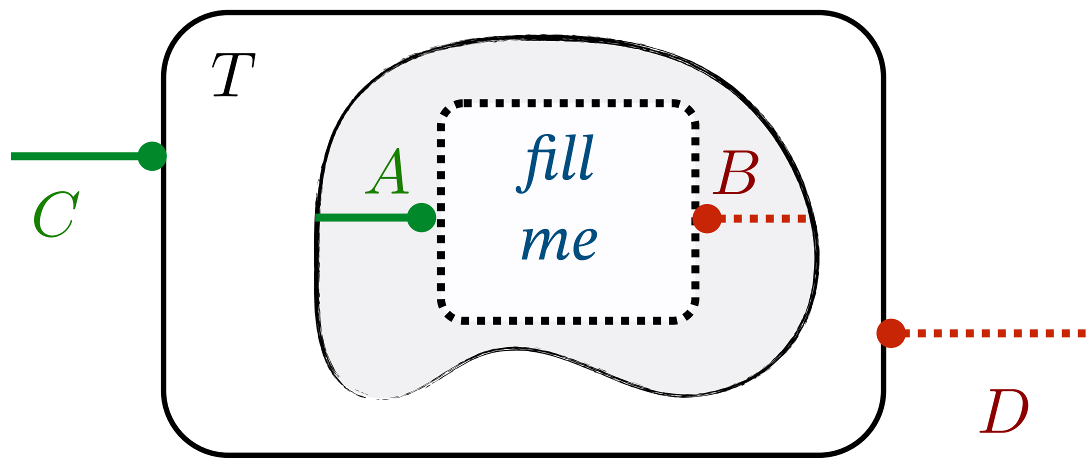
$$A^{=2} \doteq (x : A) \times (y : A) \times (p : x =_A y)$$ *two certifiably equal As*

- Interestingly, $(A_{\neq m})^{=n} \simeq (A^{=n})_{\neq m} \doteq A^{=n}_{\neq m}$

$(A_{\neq 2})^{=3}$   $(A^{=3})_{\neq 2}$

3. Higher-order structure

- Define a **“template” as a diagram with holes** that we can fill with DPs to obtain another DP.



$$T : \mathrm{Hom}(A; B) \to_{\mathbf{Pos}} \mathrm{Hom}(C; D)$$

- One can prove the following **“representation result”**:
  - For every template, the operation that fills the holes is a monotone map between DPs hom-sets.

    and viceversa
  - Every monotone map between DPs hom-sets can be expressed as a template-filling operation.

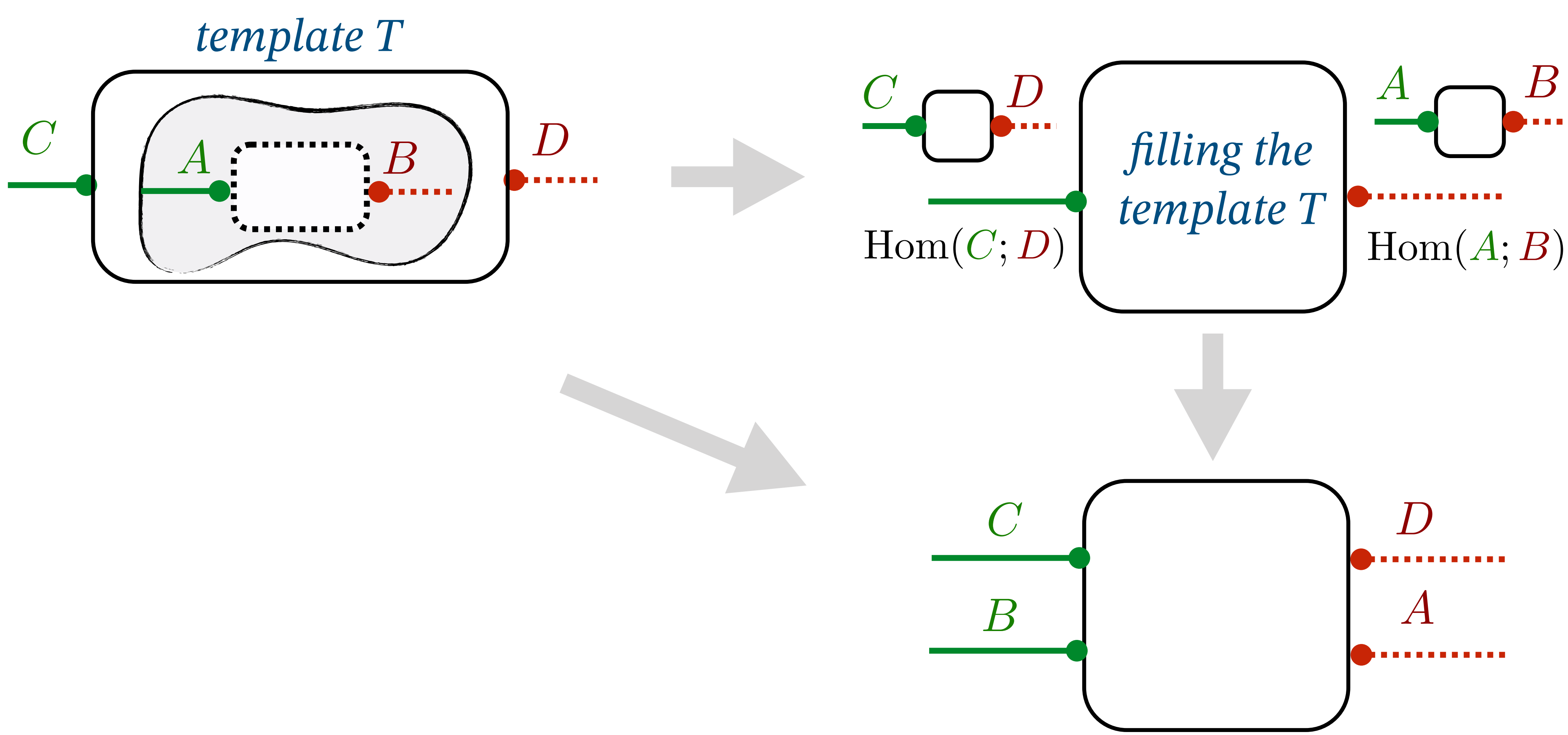
- For a $T : \mathrm{Hom}(A; B) \to_{\mathbf{Pos}} \mathrm{Hom}(A; B)$ what is $\ldots T(T(T(T(T(\perp))))) \cdots$?

3. Higher-order structure

- DP as defined is **compact closed.**

$$A \multimap (B \multimap C) \;\simeq\; (A \otimes B) \multimap C$$
$$(A \multimap B) \multimap C \;\simeq\; B \multimap (A \otimes C)$$

- This implies that **all higher-order structure collapses.**



4. Linear resources

- Linear logic is a **“logic of resources**.”
- Atoms are **uncopiable**, **unshareable**, **undiscardable** “resources”.

classical logic
(programming, etc.)

$$\frac{A \qquad A \Rightarrow B}{A \qquad A \Rightarrow B \qquad B}$$

“You are strolling in a garden finding landmarks and maps to find other landmarks.”

linear logic

$$\frac{A \qquad A \multimap B}{B}$$

“You are consuming resources”

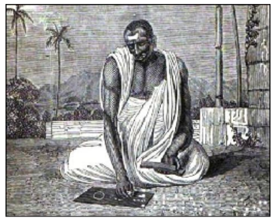
$A \multimap B$ *the type of single-use machines that eat A, produce B, and then disappear*

4. Linear resources

- Linear logic is a **“logic of resources**.”
- Atoms are **uncopiable**, **unshareable**, **undiscardable** “resources”.

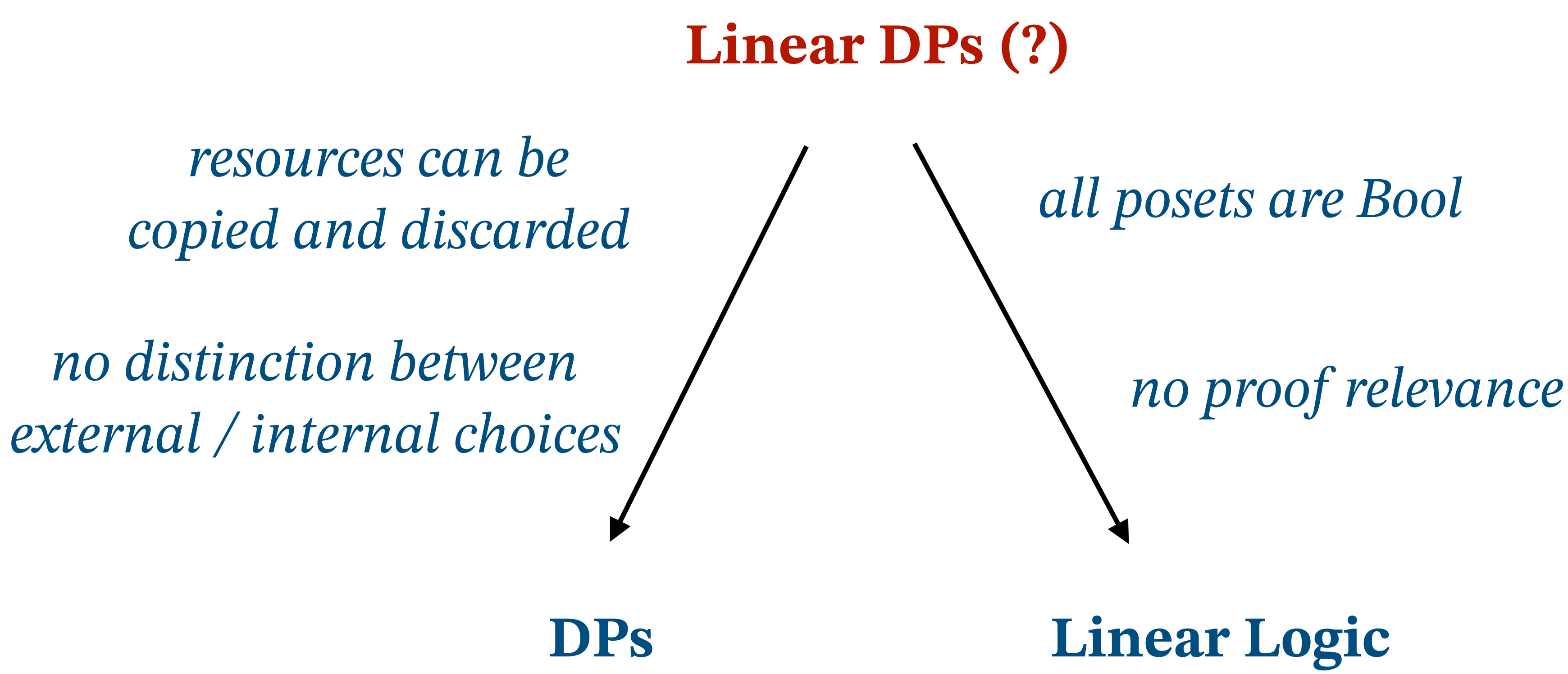
- Two “multiplicatives”, two “additives” four units, an involution, two modalities!

		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  $A^{\perp}$

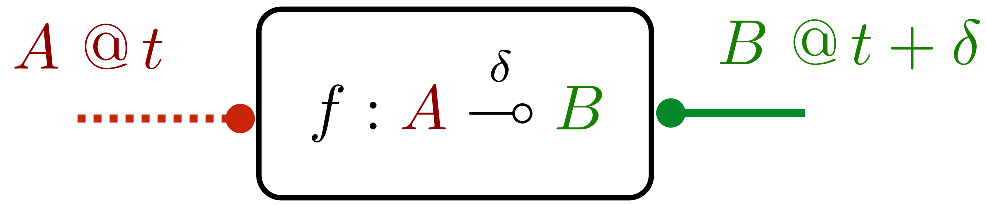
4. Linear resources

- What would **a theory of linear co-design** be?



5. Temporal resources

- Suppose **I want to keep track of the time it takes to produce something.**



- Let's decorate the morphisms and **refine the hom-types by time** (some commutative monoid).

$$\mathrm{id} : A \overset{0}{\multimap} A$$

$$A \overset{s}{\multimap} A \quad \text{might be uninhabited}$$

$$\text{ice} \overset{s\ ?}{\multimap} \text{ice}$$

$$\text{ice} \overset{s}{\multimap} \text{water}$$

$$\text{ice, fridge} \overset{s}{\multimap} \text{ice}$$

- **Composition** is as you expect:

$$\frac{A \overset{\alpha}{\multimap} B \qquad B \overset{\beta}{\multimap} C}{A \overset{\alpha+\beta}{\multimap} C}$$

5. Temporal resources

- **Monoidal composition** is only allowed if the temporal duration matches or the resources are "non perishable".

$$\frac{A \overset{\alpha}{\multimap} B \qquad C \overset{\alpha}{\multimap} D}{A \otimes C \overset{\alpha}{\multimap} B \otimes D}$$

- The higher order structure does not collapse anymore:

$$(A \overset{\alpha}{\multimap} B) \overset{\beta}{\multimap} C \quad \simeq^{?} \quad B \overset{\gamma}{\multimap} (A \otimes C)$$

$$A \overset{\alpha}{\multimap} (B \overset{\beta}{\multimap} C) \quad \simeq^{?} \quad (A \otimes B) \overset{\gamma}{\multimap} C$$

5. Temporal resources

- **Negative time** is also interesting
  and not quite equivalent to negative resources.

- You have promised that
  you'll give me \$10 tomorrow.

  I have: $\mathbf{1} \overset{\text{1 day}}{\multimap} \$10$

  You have: $\mathbf{1} \overset{\text{1 day}}{\multimap} -\$10$

  or: $\mathbf{1} \overset{\text{1 day}}{\multimap} (\$10 \overset{0}{\multimap} \mathbf{1})$

- **Lending:**
  If I can lend \$10 now,
  I can have \$11 back later.

  ability to lend $\$10 \overset{\text{1 week}}{\multimap} \$11$

- **Borrowing:**
  If I can repay \$11 in 1 week,
  I can have \$10 now.

  ability to borrow $\$11 \overset{\text{-1 week}}{\multimap} \$10$

- Net present value axioms
  with interest rate α :

  access to credit $\multimap$

$$\mathbf{1} \overset{t}{\multimap} \$x \quad \circ\!\!-\!\!\circ \quad \mathbf{1} \overset{t+\delta}{\multimap} \alpha^{\delta} \cdot \$x$$

6. Spatial resources

- We can describe **constraints on the arrangements of resources.**
- Take a simplified metric structure where things are either "close" or "far apart".

$A \wr B$ Two **separable resources.**

$A \asymp B$ Two **inseparable resources.**

$A \leftrightsquigarrow B$ Two resources that are **constrained to be apart.**

- It's a **linear distributive category twice over** (3 monoidal structures):

$$\begin{tikzcd} & \wr \arrow[dl] \arrow[dr] & \\ \asymp & & \leftrightsquigarrow \end{tikzcd}$$

tensor strengths

strength morphisms

$$A \wr (B \asymp C) \to (A \wr B) \asymp C$$

$$A \wr (B \leftrightsquigarrow C) \to (A \wr B) \leftrightsquigarrow C$$

- More generally, given a metric space we can create an infinite amount of tensor operations representing resource placement constraints.
- (If object = tensors, morphisms = tensor strength domination, what does this category look like?)

$$A \overset{M}{\underset{m}{\square}} B$$

You must place A and B in sets $S_A, S_B \subset \mathcal{S}$ such that $m \leq d_{\mathcal{S}}(S_A, S_B) \leq M$.

6. Spatial resources

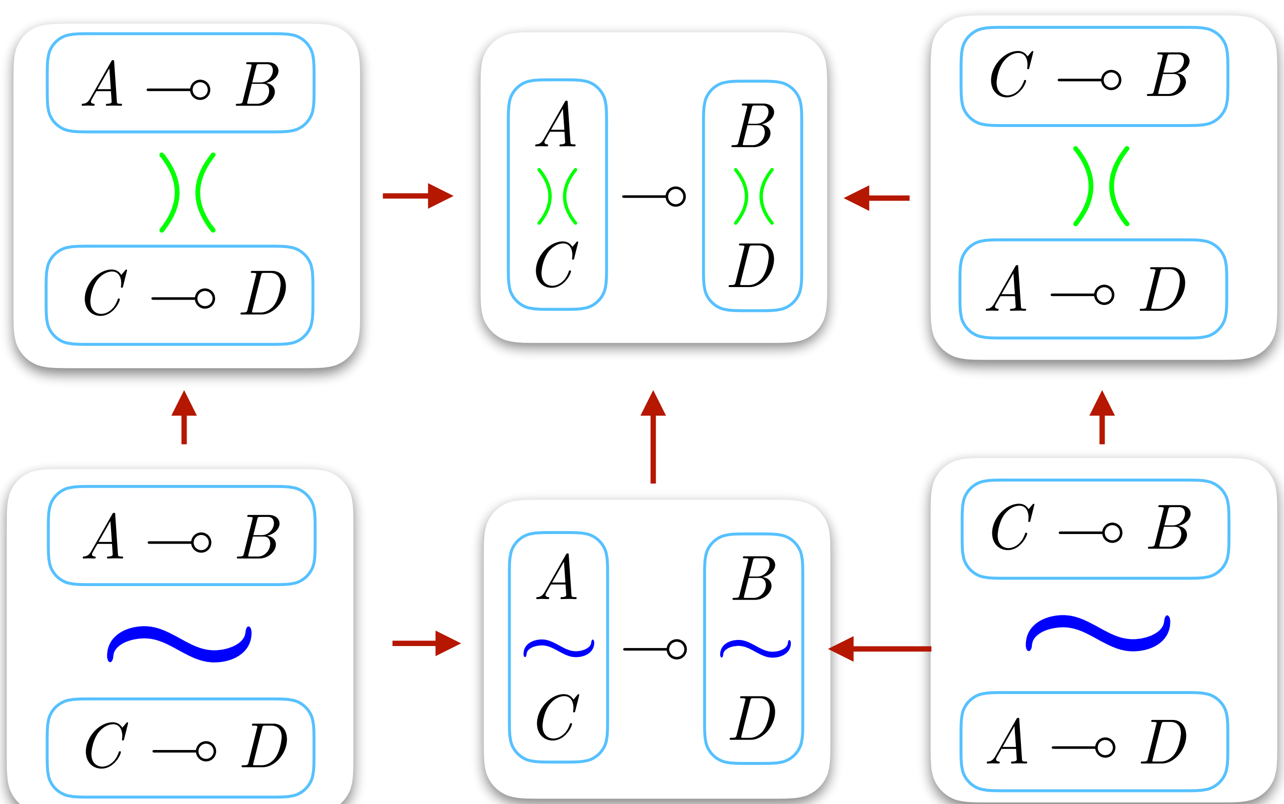
- Take a simplified metric structure where things are either “close” or “far apart”.

$A \wr B$	Two **separable resources.**
$A \asymp B$	Two **inseparable resources.**
$A \leftrightsquigarrow B$	Two resources that are **constrained to be apart.**

- Other constraints: e.g. **machines need to be close to materials** for the production to happen.

$$\frac{A \asymp (A \multimap B)}{B}$$

- Inspires to be a bit **more “precise” in “subtyping”**:

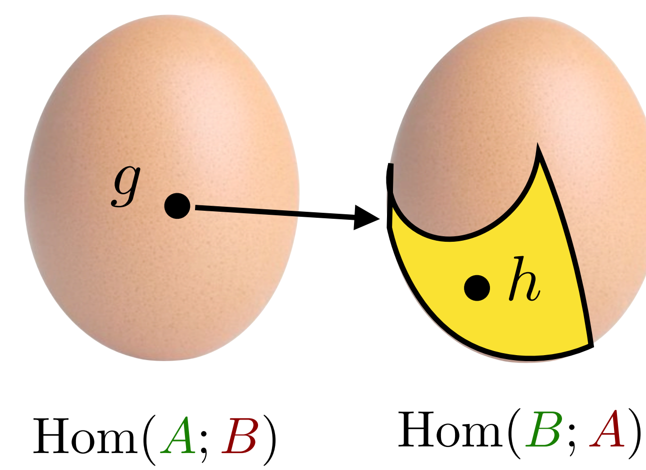


7. Negative co-design

- In this universe there is conservation of mass and energy.

> **The real world axiom.**
>
> For any $f : A \nrightarrow A$ necessarily $f \leq \mathrm{Id}_A$

- This means that if you give me any profunctor $g : A \nrightarrow B$
  - **You are telling me something positive** about what B I can produce from A.
  - And, **you are telling me something negative** about what A I can produce from B.



For any $h : B \nrightarrow A$ necessarily $(g\,;\,h) \leq \mathrm{Id}_A$

$(h\,;\,g) \leq \mathrm{Id}_B$

lower sets on $\mathrm{Hom}(B; A)$

- Gives an entire new symmetry to explore!
  Very relevant to the computation side (impossibility results).

Conclusions