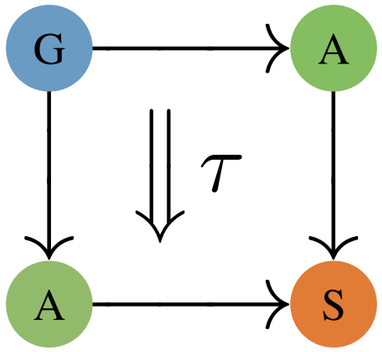
Applied Category Theory for Dynamics and Control - Introduction

7/7/2025

James Fairbanks



Spectrum of Scientific Computing Technology

Programs that have explicit representations of the model

Arbitrary Code

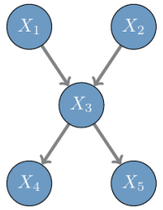
Domain Specific Languages

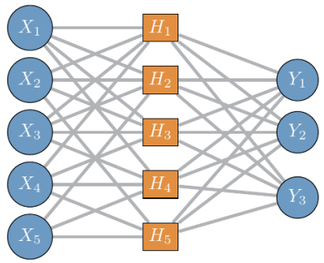
Modeling Frameworks

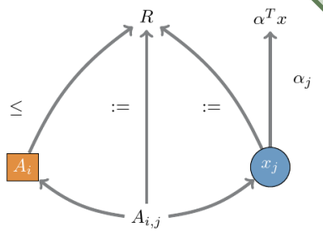
Computer Algebra Systems

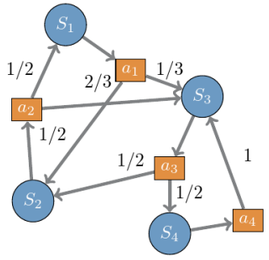
SymPy

Vision: Model Aware Scientific Computing with Categories

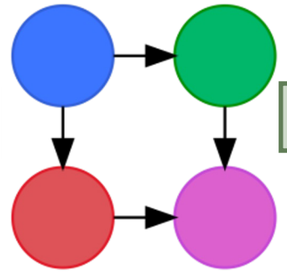




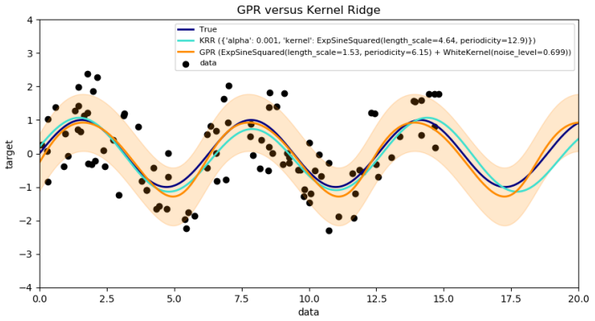


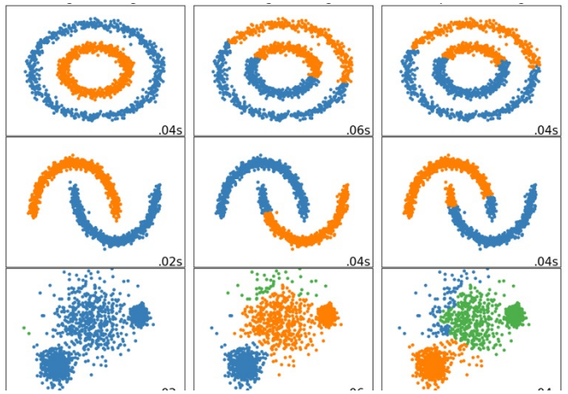


AlgebraicJulia

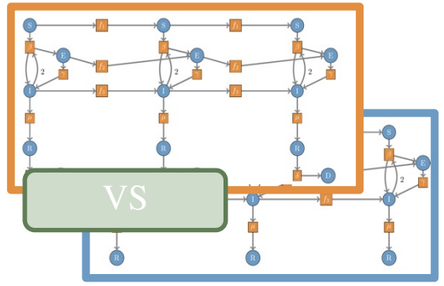


Model Specific Tools for Scientists



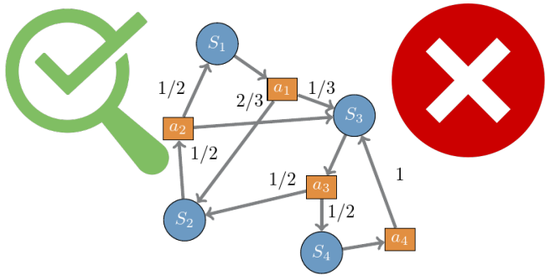


Calibration



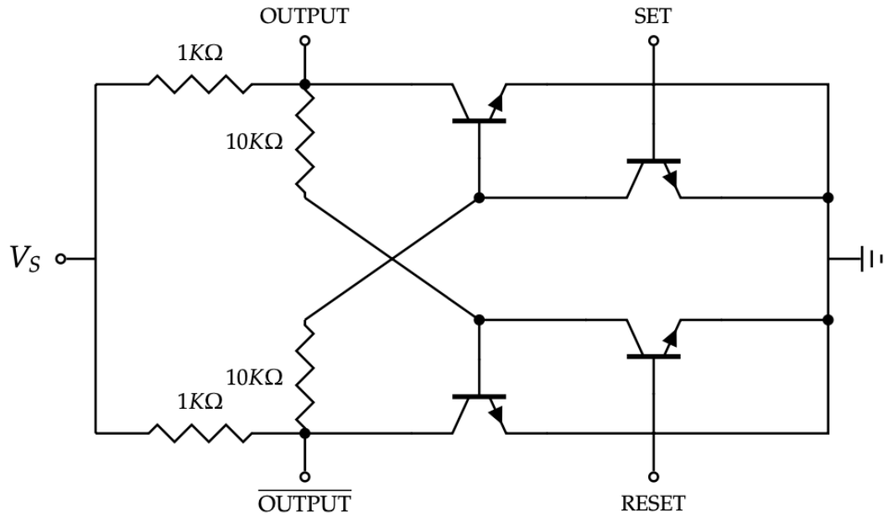
Comparison

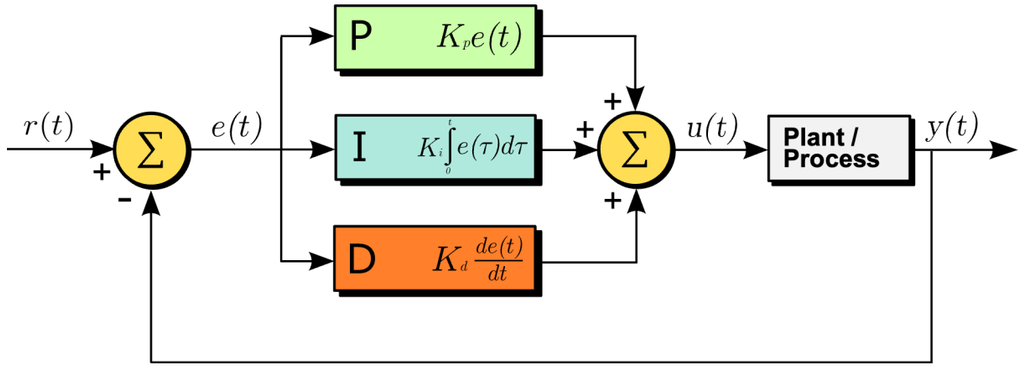
Selection

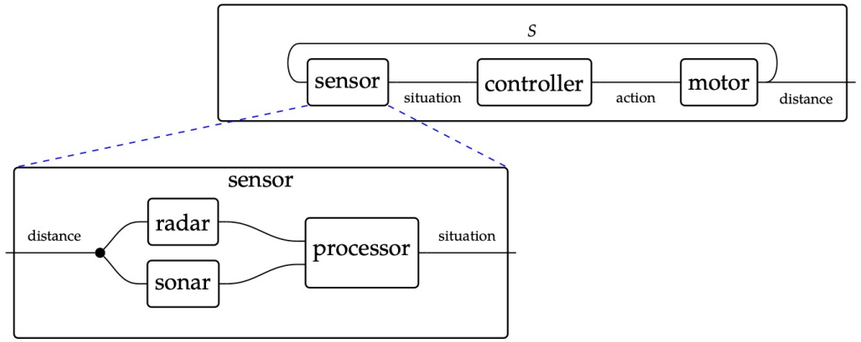


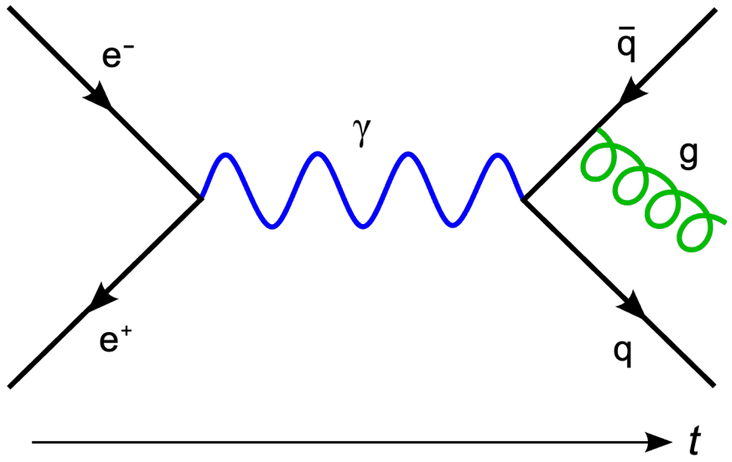
Verification & Validation

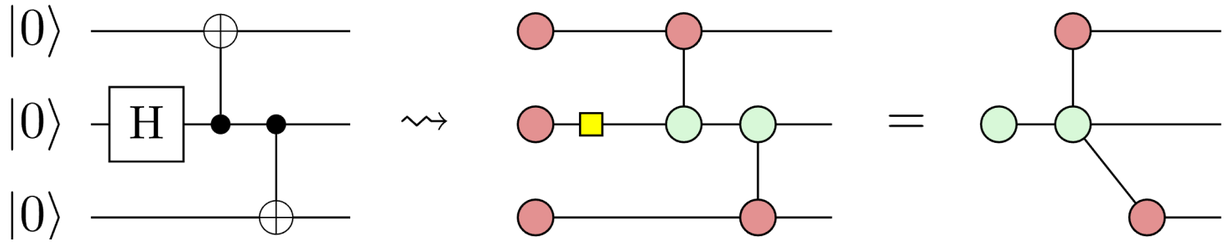
Formal Scientific Diagrams







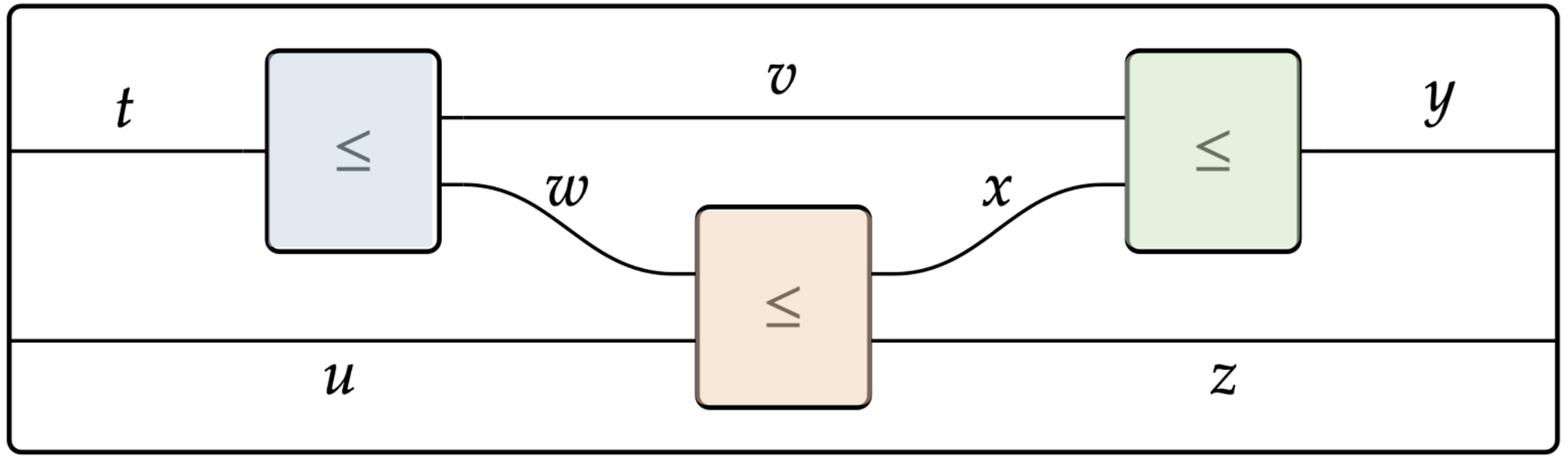




Formal Proofs with Inequalities

- Hypotheses: $t \leq v + w$ $w + u \leq x + z$ $v + x \leq y$

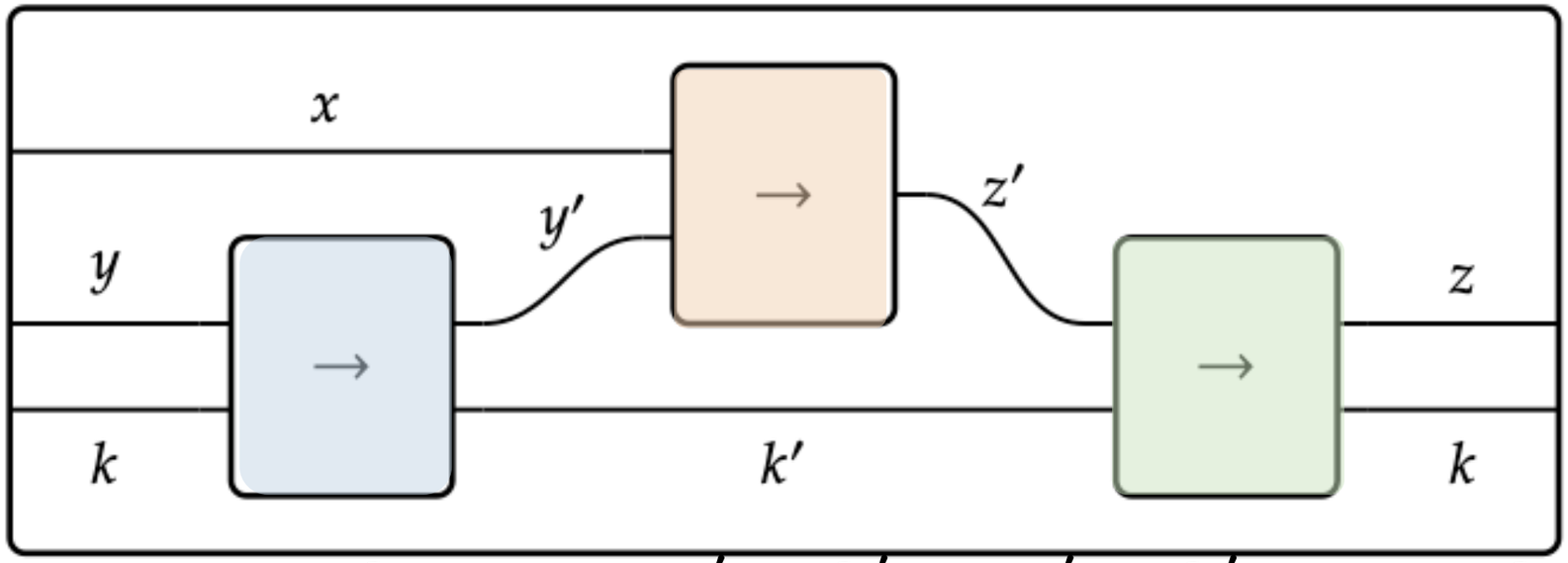
- Proof:



- Conclusion: $t + u \leq y + z$

Automated Reasoning in Chemical Systems

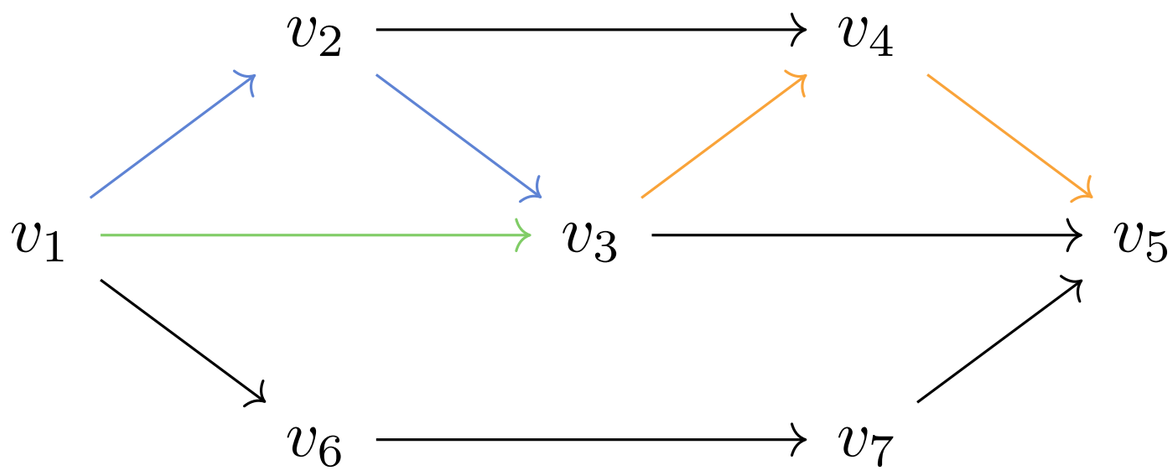
- Primitive Reactions: $y + k \rightarrow y' + k'$ $x + y' \rightarrow z'$ $z' + k' \rightarrow z + k$
- Chemical Process:



$$x + y + k \rightarrow x + y' + k' \rightarrow z' + k' \rightarrow z + k$$

- Composite Reaction: $\boxed{x + y + k \rightarrow z + k}$

Paths in a Graph are Free Categories



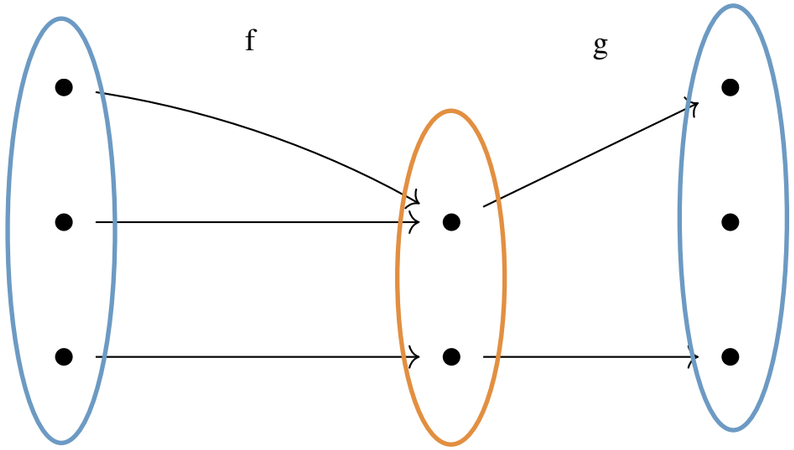
$$v_1v_2v_3 \cdot v_3v_4v_5 = v_1v_2v_3v_4v_5$$

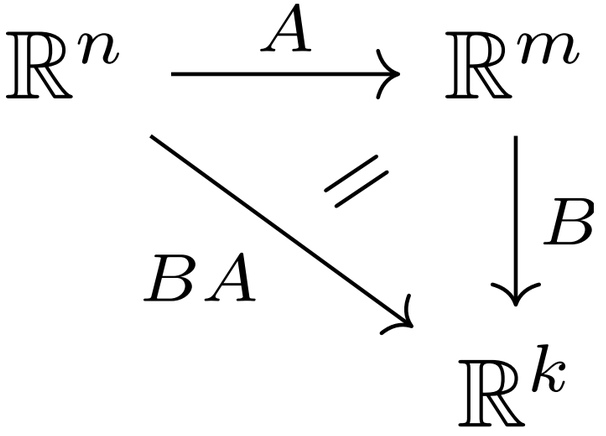
$$v_1v_3 \cdot v_3v_4v_5 = v_1v_3v_4v_5$$

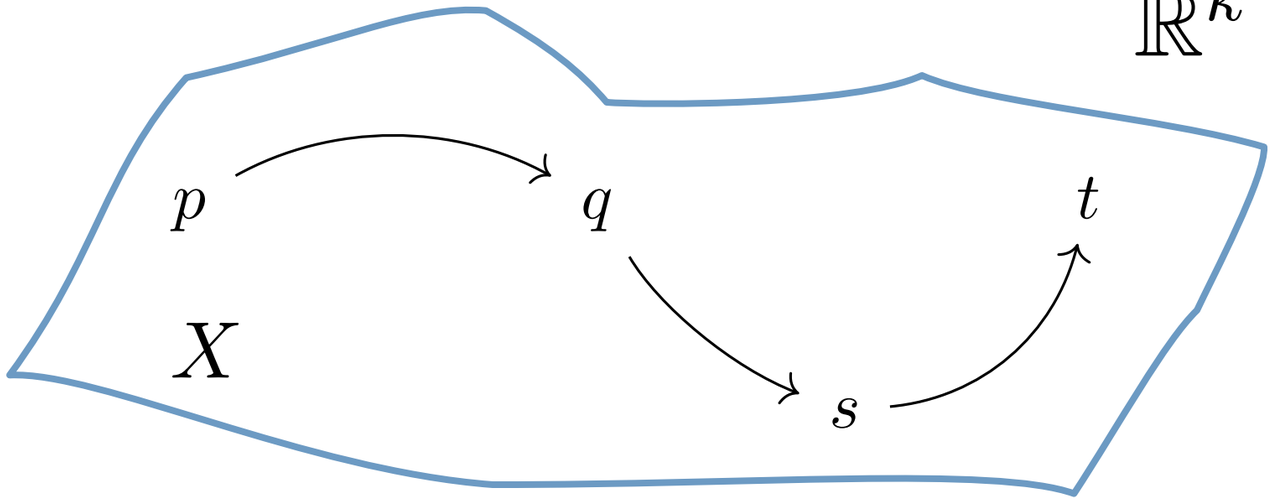
- Every graph presents a free category
- Paths can only compose if they share an endpoint.
- Graph theory studies these categories without the language of CT

Examples

- Sets and Functions
- Finite Sets and Functions
- Vector Spaces and Linear Maps (Matrices)
- Topological Spaces and Continuous Functions
- Convex Spaces and Convex Functions
- Points in a Space and Paths in that Space
- Dynamical Systems and Changes of Coordinates







Functors

- Every mathematical structure has “structure preserving maps”
- For Categories, these maps are *Functors*
- Map Objects of C to Objects of D and Arrows in C to Arrows in D, such that composition and identities are preserved

$$F_0 : Ob_C \to Ob_D$$

$$F_1 : Hom_C \to Hom_D$$

$$F_1(id(A)) = id(F_0(A))$$

$$F_1(f \circ_C g) = F_1(f) \circ_D F_1(g)$$

Example: The free vector space on a set

$$V : Set \to Vect$$

$$n \mapsto \mathbb{R}^n$$

$$f : n \to m \mapsto A_f : \mathbb{R}^n \to \mathbb{R}^m$$

$$A_f\,(j,i) = \begin{cases} 1 \text{ if } f(i) = j \\ 0 \text{ else} \end{cases}$$

Functorial Semantics in Programming Languages

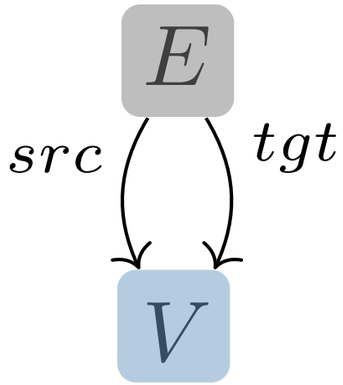
$$\text{Syntax} \xrightarrow{\text{Compile}} \text{Semantics}$$

- Syntax is combinatorial trees, graphs, hypergraphs
- Semantics is quantitative, functions, relations, shapes, distributions
- Functor from syntax to semantics recursively generates programs
- Structure of the categories/functor determines allowable language operations

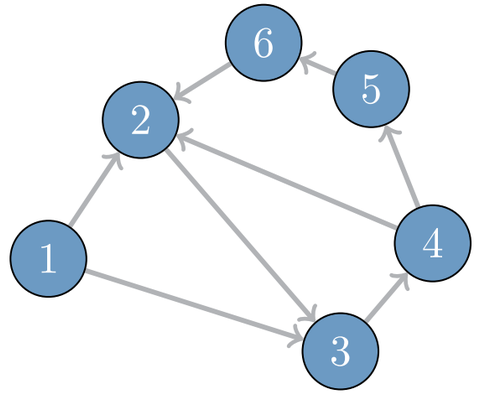
Attributed $\mathcal{C}$-Sets: Categorical Data Structures

Graphs are ubiquitous because they a simple & useful structure

Gr

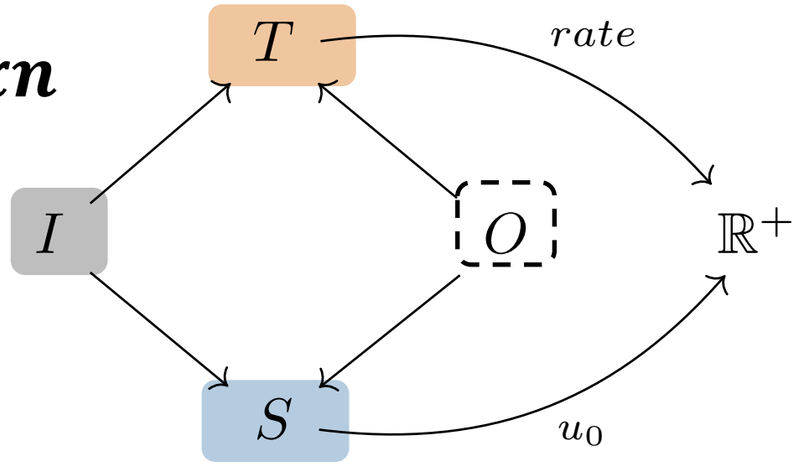


Gr − Set

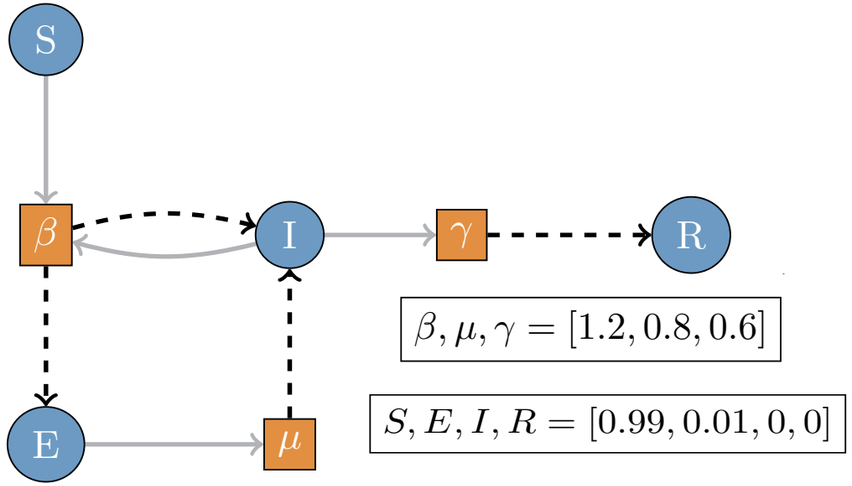


Attributed $\mathcal{C}$-Sets generalize algebraic graph theory

Rxn

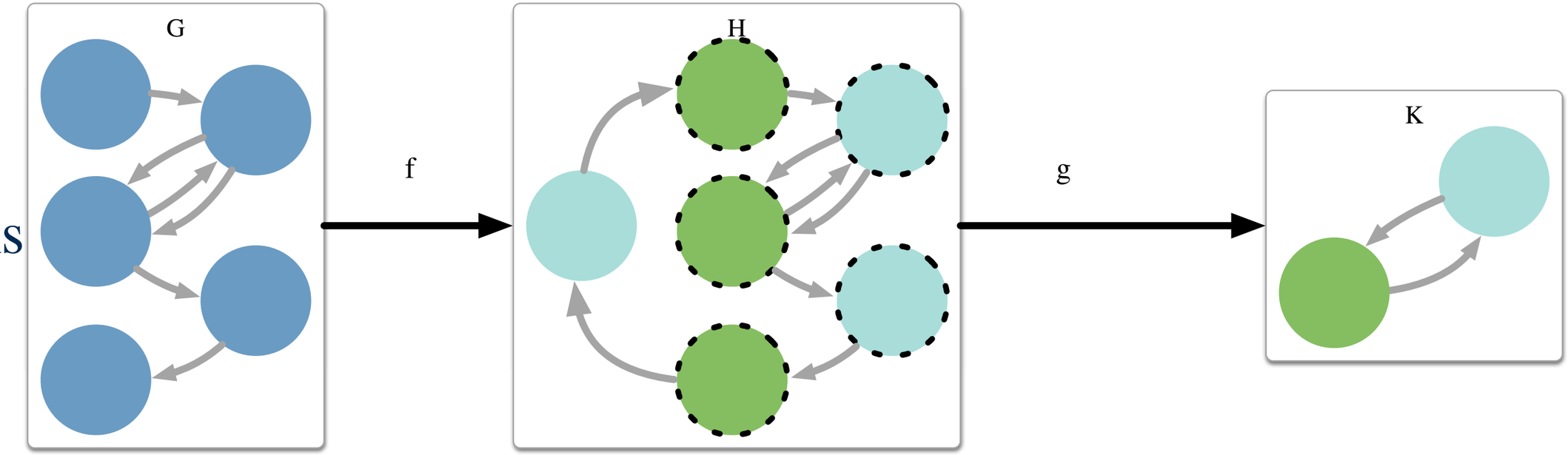


Rxn − Set

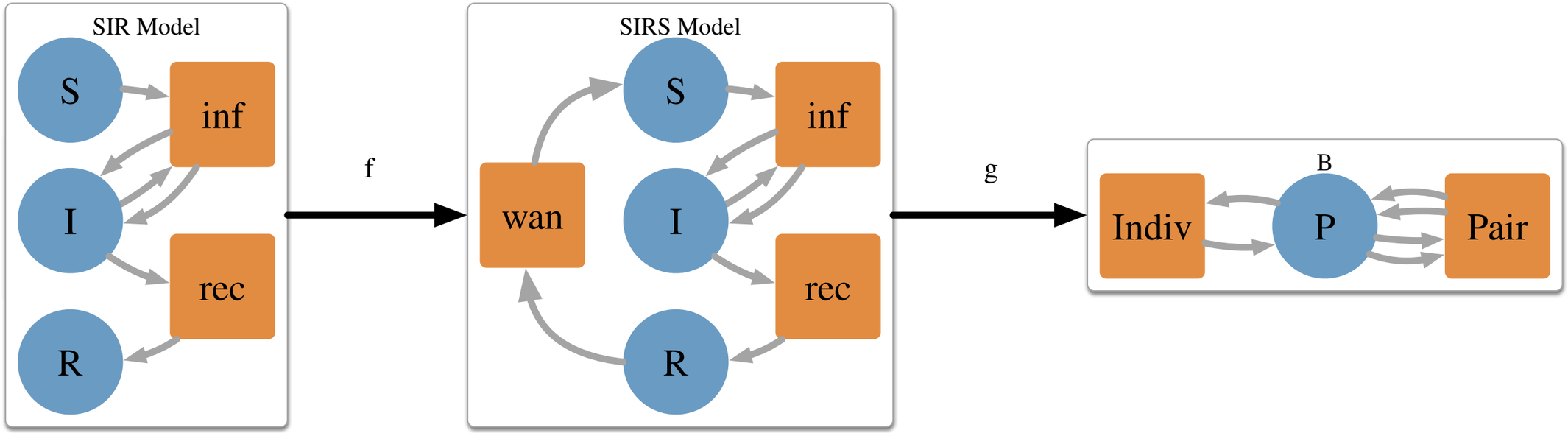


Attributed $\mathcal{C}$-Sets: Categorical Data Structures

Graph Morphisms

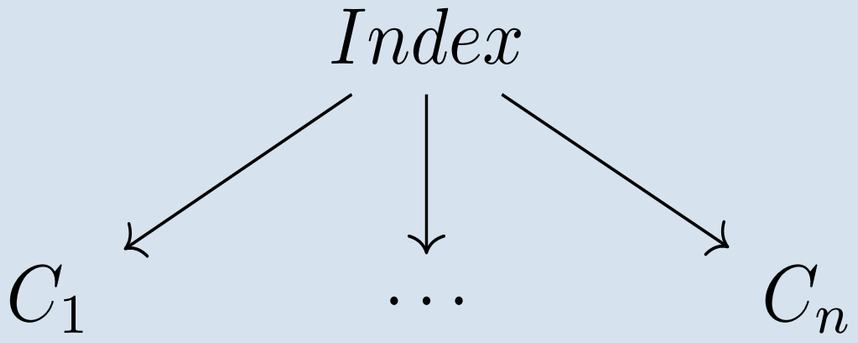


Petri Net Morphisms

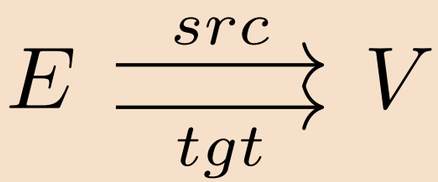


ACSets generalize Tables and Graphs

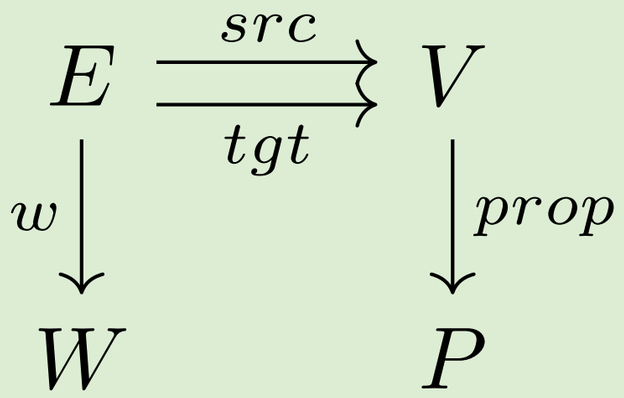
Table



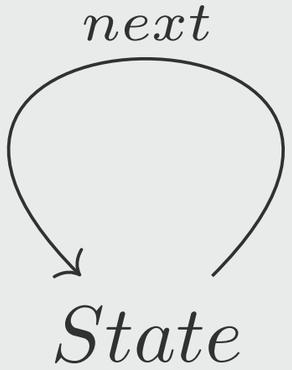
Graph



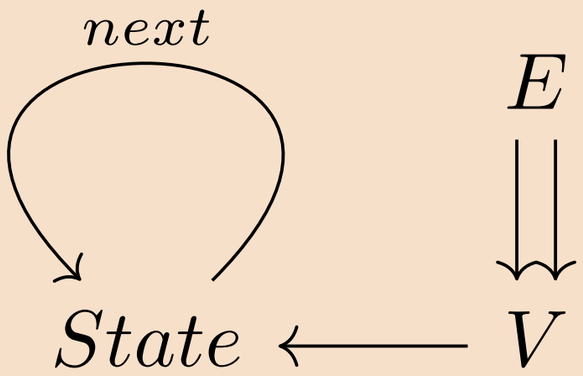
Weighted, Property Graph



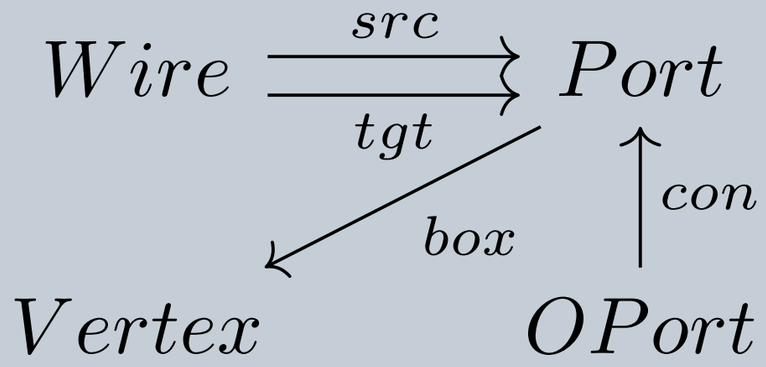
Discrete Dynamical System



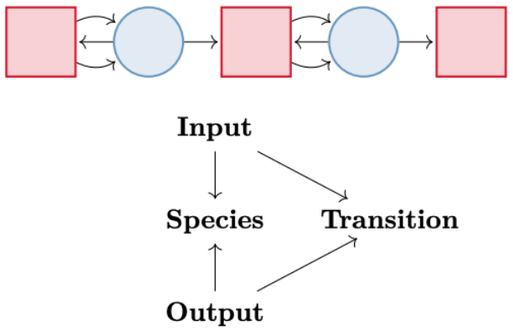
DDS on a Graph



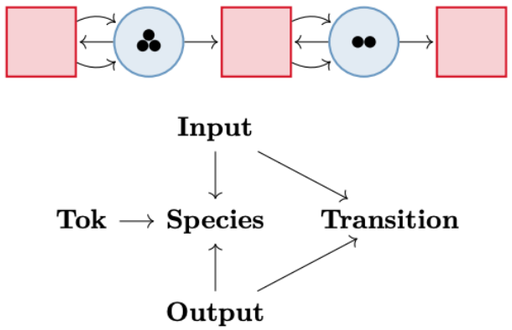
Open Circular Port Graphs



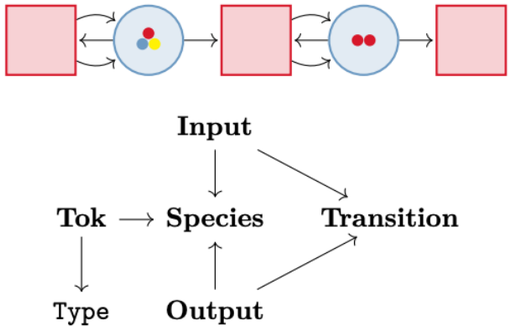
Building Rich Data Structures as ACSets



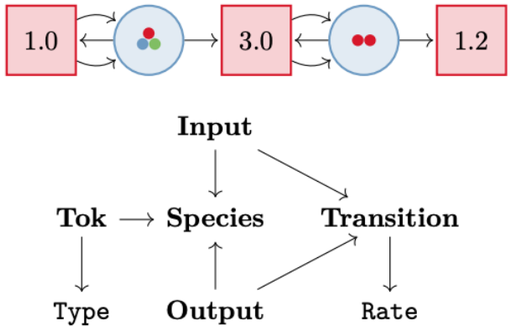
(a) Petri net



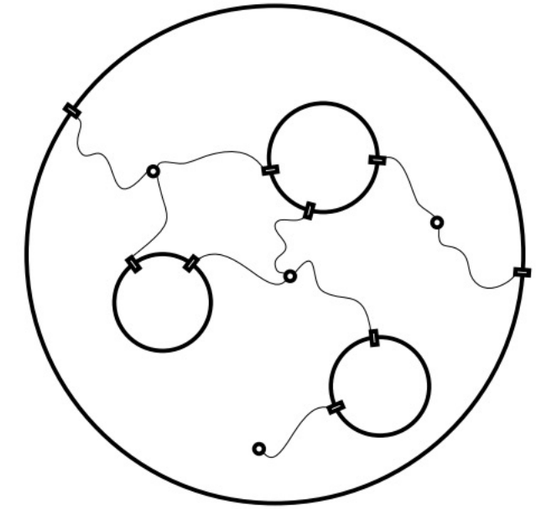
(b) Petri net with tokens

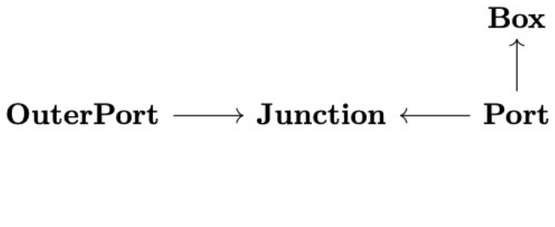


(c) Petri net with typed tokens

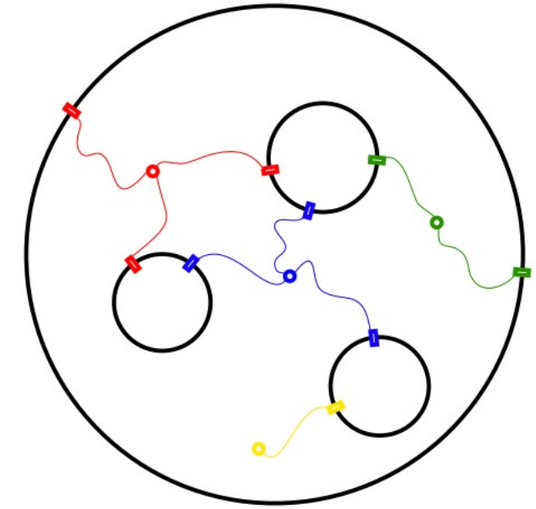


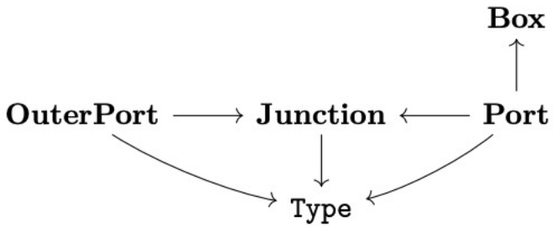
(d) Petri net with typed tokens and rates





(a) Undirected wiring diagram



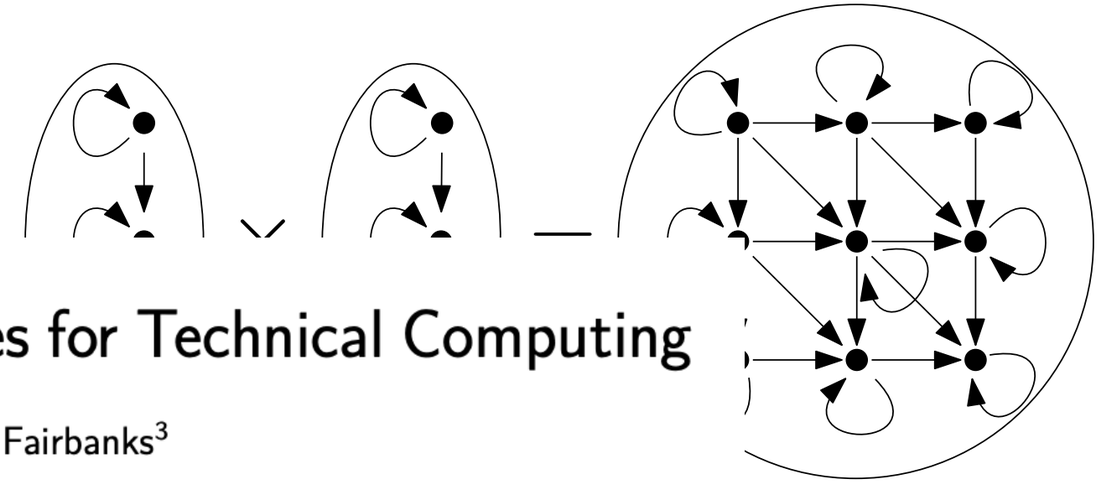


(b) Typed undirected wiring diagram

- The data of a mathematical model can usually be stored as an ACSet
- Highly generic code implies low complexity cost for adding features

ACSets have many nice features

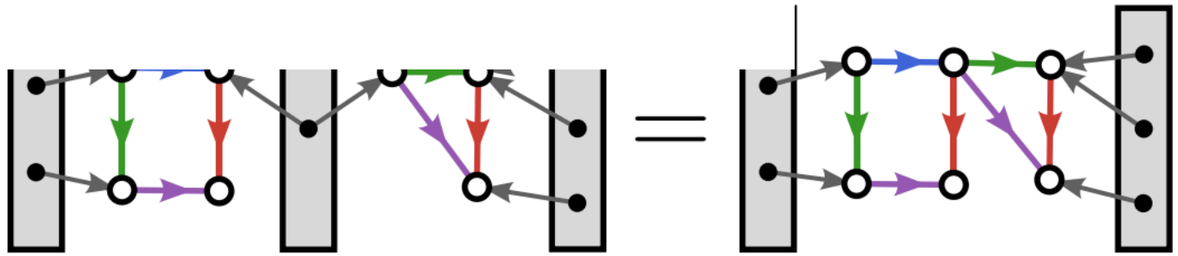
- All Finite Limits:
  - An initial (singleton) object
  - Products (products)
  - Pullbacks
- All Finite C
  - A termina
  - Coproducts (disjoint unions)
  - Pushouts (unions)
- Limits and Colimits have point-wise formulas



Categorical Data Structures for Technical Computing

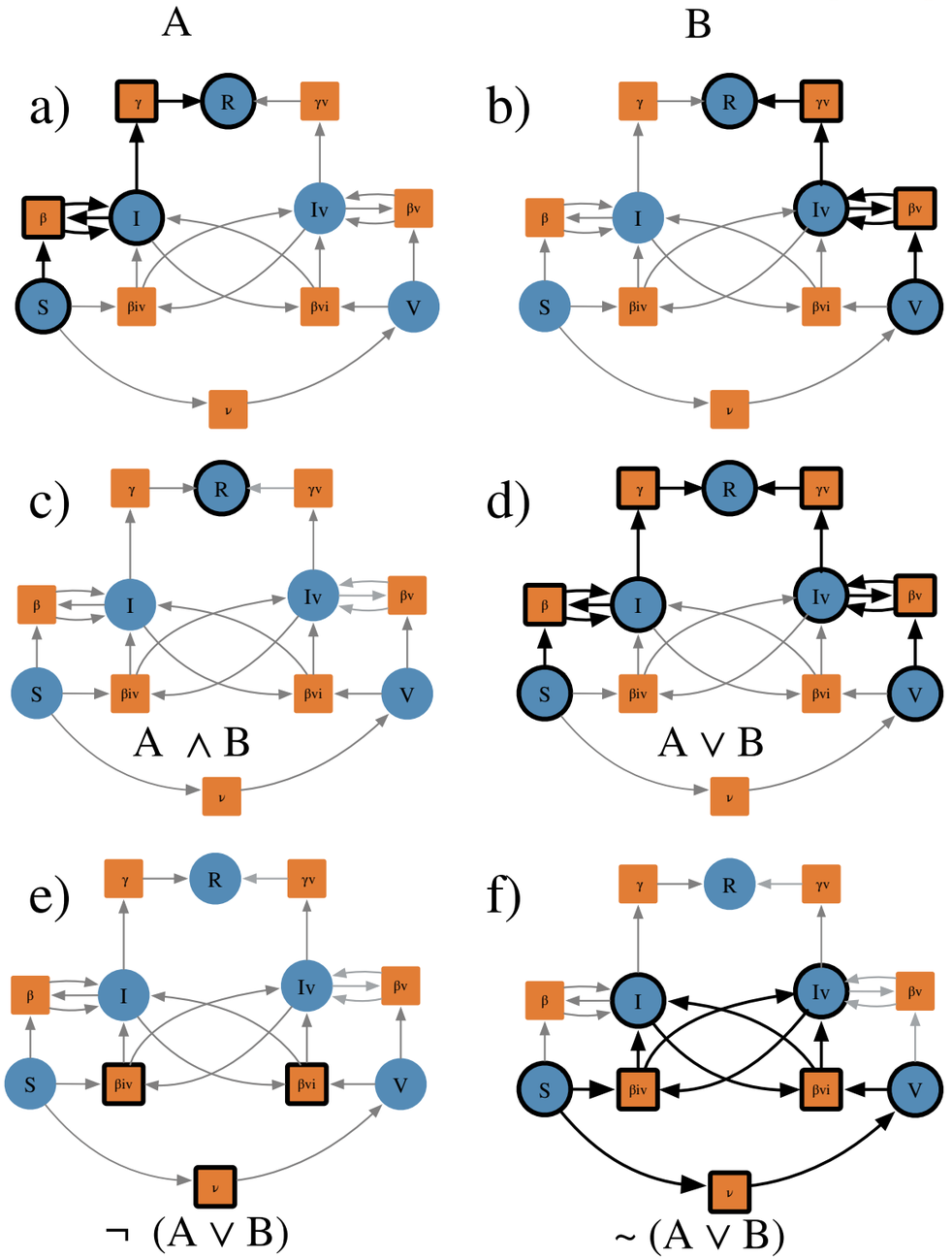
Evan Patterson[1], Owen Lynch[2], and James Fairbanks[3]

[1]Topos Institute, California, USA
[2]Universiteit Utrecht, Mathematics Department, Utrecht, The Netherlands
[3]University of Florida, Computer & Information Science & Engineering, Florida, USA



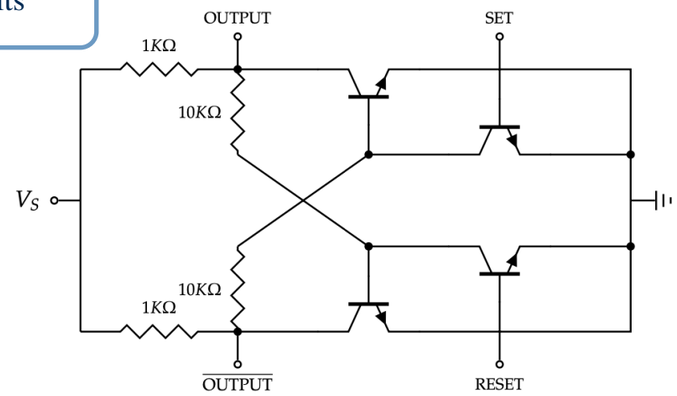
ACSets are Combinatorial, Logical, and Topological

- A C-Set is a bi-Heyting Topos
- For a fixed object, you look at all the subobjects
- They form a lattice with intersection and union
- Logic is intuitionistic because LEM fails.
- Two types of negation
- Combinatorial Boundary Operator

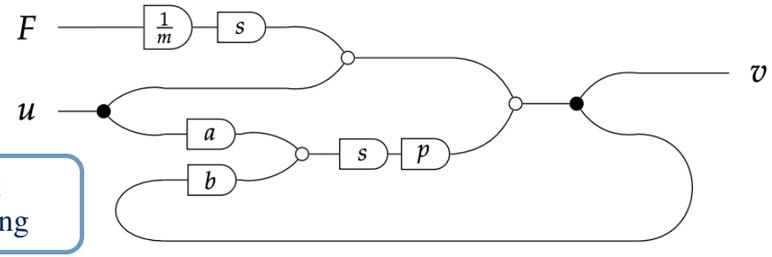


Wiring Diagrams are ACSets

Circuits



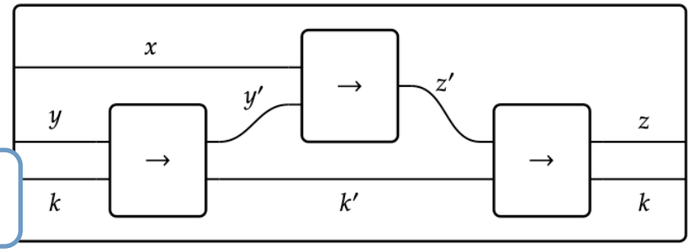
Signal Processing



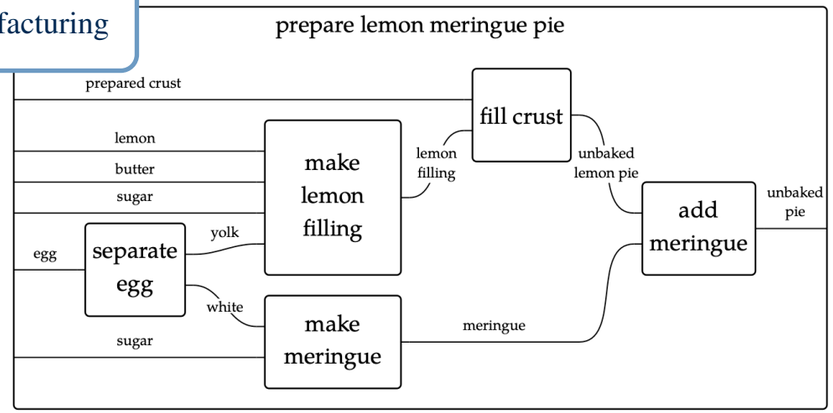
$$v = \int \frac{1}{m} F(t)dt + u(t) + p \int au(t) + bv(t)dt.$$

Chemistry

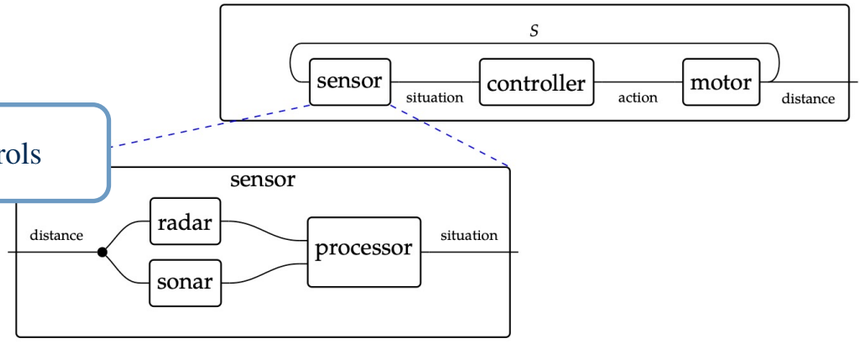
$$y + k \to y' + k' \qquad x + y' \to z' \qquad z' + k' \to z + k$$



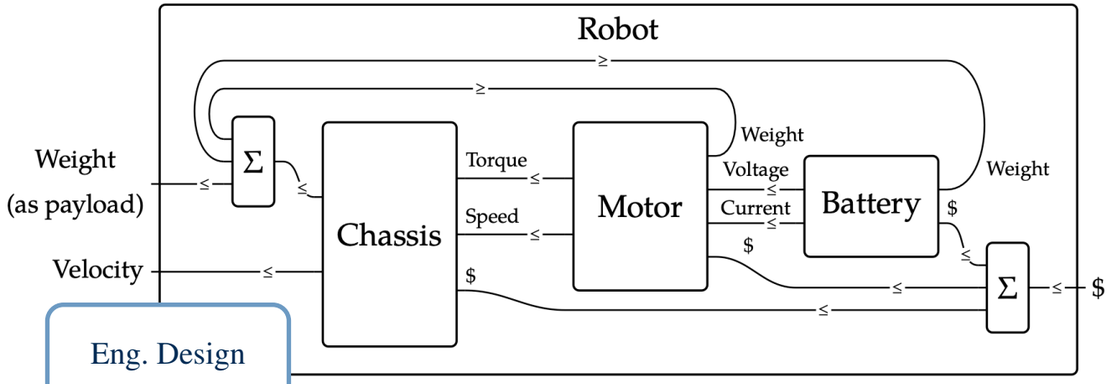
Manufacturing



Controls



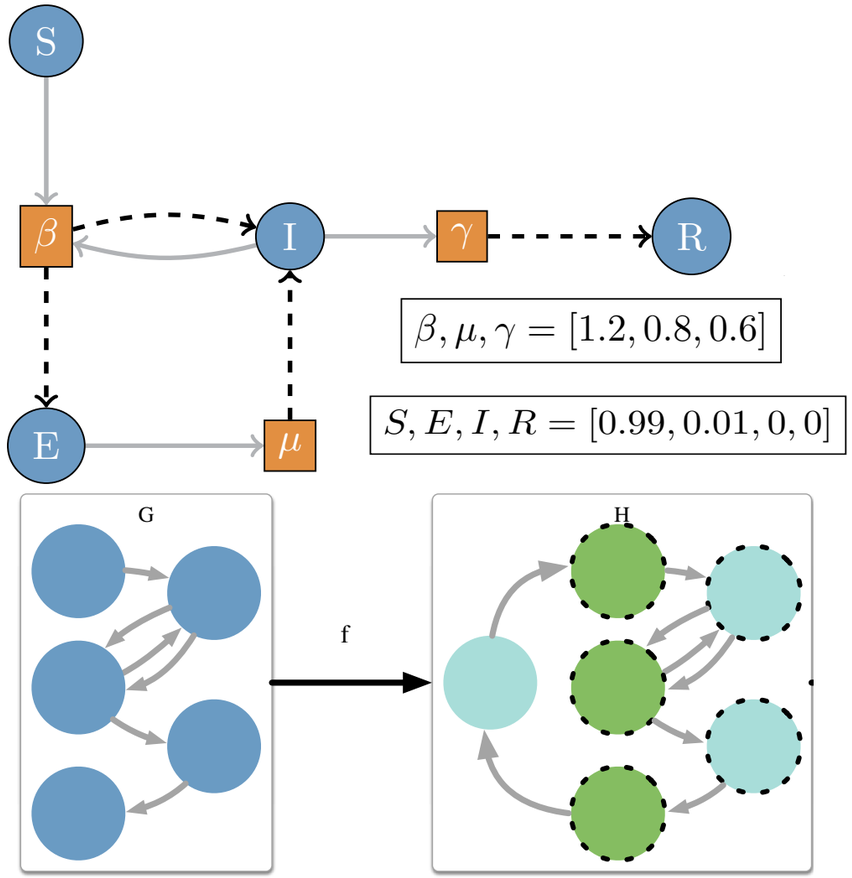
Eng. Design



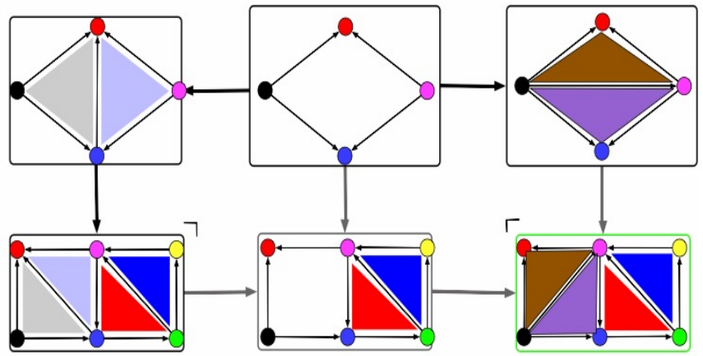
Figures from Fong and Spivak 2019

Languages, Data Structures, Algorithms

- Developed infrastructure for software based on Category Theory
- Our generic algorithms are faster than specialized code in SOTA
- Proves that Generality vs Performance tradeoff typical to HPC codes is surmountable
- Data Structures Paper *Compositionality,* premier journal of the Applied Category Theory field.
- Model Structure Modification paper Accepted to *Int. Conf. Graph Transformation* Apr 14th, 2022 (best paper award)



Benchmarks of Catlab vs Graphs.jl (SOTA graph library) showed speedup in 16 of 28 benchmarks. Performance within 2X of SOTA on 10 of 28 benchmarks.



Mesh size	Catlab (s)	ReGraph (s)
2 by 2	1.2×10^{-4}	5.3×10^{-3}
2 by 3	2.7×10^{-4}	8.0
2 by 4	4.7×10^{-4}	1313.3
2 by 5	6.7×10^{-4}	44979.8

Benchmark of Catlab vs ReGraph (a SOTA graph rewriting library). YFA developed software is asymptotically more efficient, achieving over 8 orders of magnitude higher performance on a small problem instance.