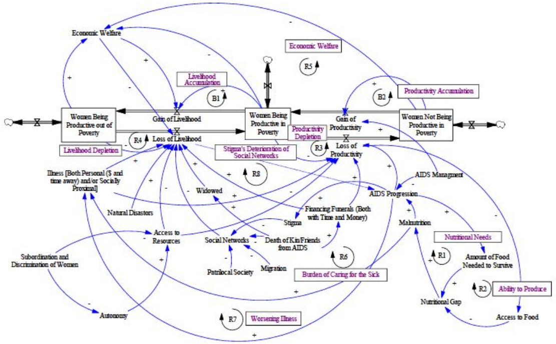
Compositional SD: The Higher Mathematics Underlying System Dynamics Diagrams & Practice

Xiaoyan Li, Nathaniel Osgood

University of Saskatchewan

Joint work with John Baez, Evan Patterson, Sophie Libkind, Alex Alegre, Eric Redekopp, Thomas Purdy, Nicholas Meadows

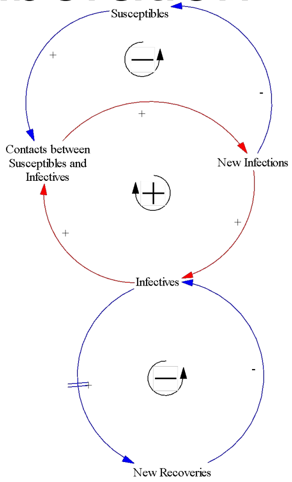
Three Tools: Successive Diagrammatic Elaboration

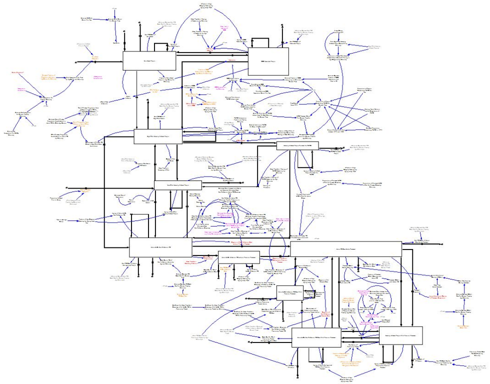


Causal Loop Diagrams

System Structure Diagrams

Stock & Flow Diagrams





2. Motivations: Some Gaps in Theory of SD Models

- Absence of mathematical characterization of diagrams
- Inability to formally relate
  - One instance of a diagram type (e.g., CLD, Sys. Structure) to another of the same type
  - Instances of distinct diagram types
- No mathematical framework for types of model composition
- Lack of capacity to mathematically specify reusable higher levels of abstraction (molecules)
- Manual stratification obscures dynamic pathways
- Semantic inflexibility: privileging ODE interpretation

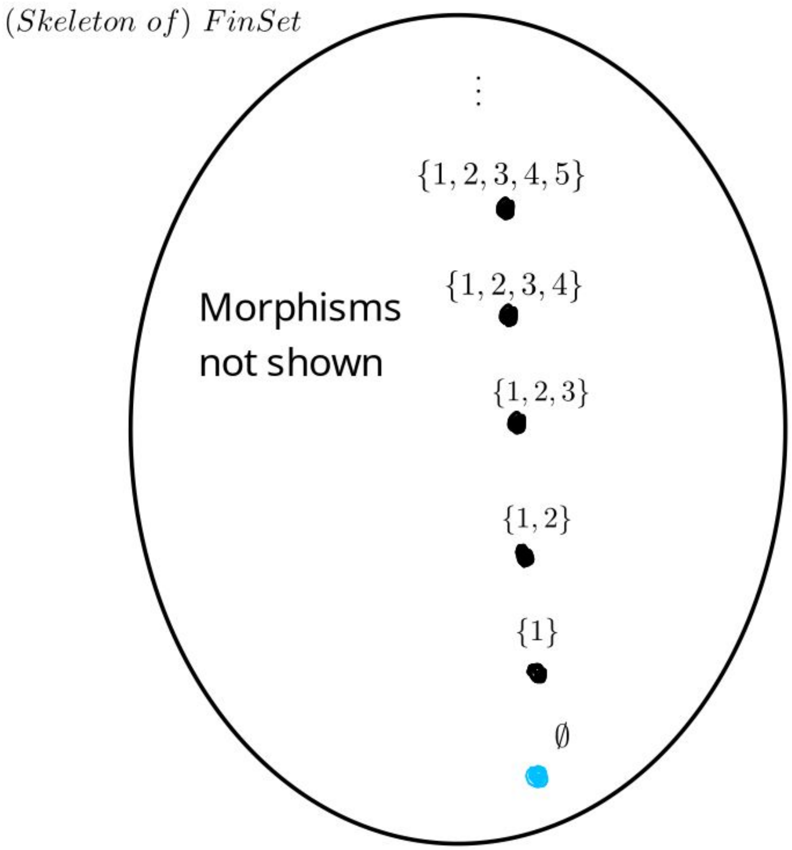
Why a Categorical Approach?

- Transparency for stakeholders
- Support for diagrammatic reasoning
- Modularity
- Abstraction
- Composability
- Separation of syntax & semantics: Capacity to express diverse semantic domains (simulation, calibration, loop gain, eigenvalue elastic., computational statistics, …)
- Mapping between diagram types
- Support for provably safe migration of representation with schema evolution
- In hybrid context, ready support for multiscale modeling
- Analysis & safe transformations for optimization & parallelization

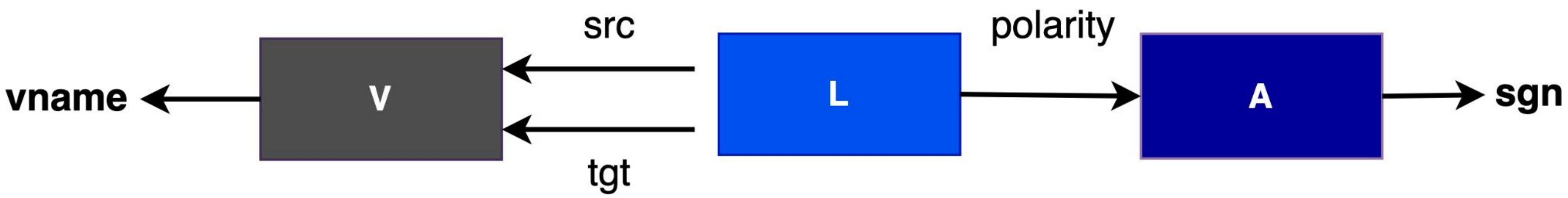
Key Elements of Approach

- Encoding of each diagram type using ACSets (attributed copresheaves) on respective schemas
- Domain specific languages
- Open diagrams composition via pushouts of undirected wiring diagrams encoded as multicospans
- Model stratification via pullbacks over type diagrams
- Functorial semantics: Mapping ACSet into semantic domain
- Functors mapping diagrams with greater structure to those with lower structure
- Support for programmatic modeling in AlgebraicJulia
- Browser-based support for real-time collaborative model building, manipulation & interpretation absent categorical knowledge

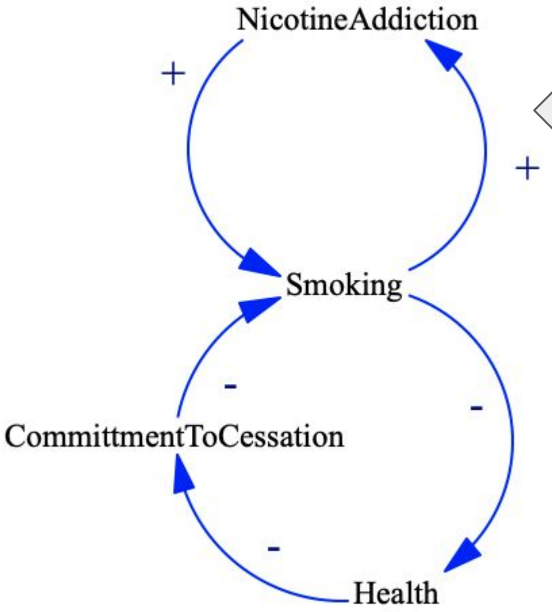
The Category FinSet



Category X of Causal Loop Diagrams: FinSet^{SchCLD}

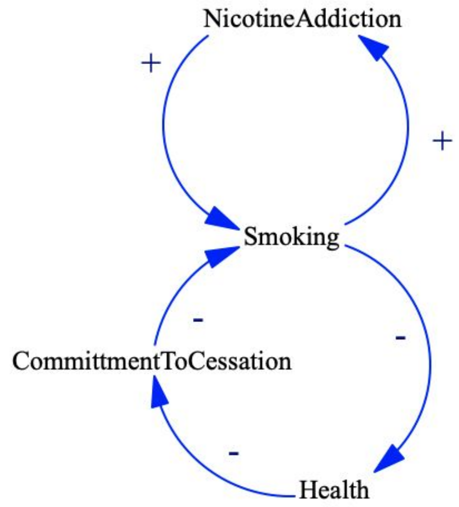


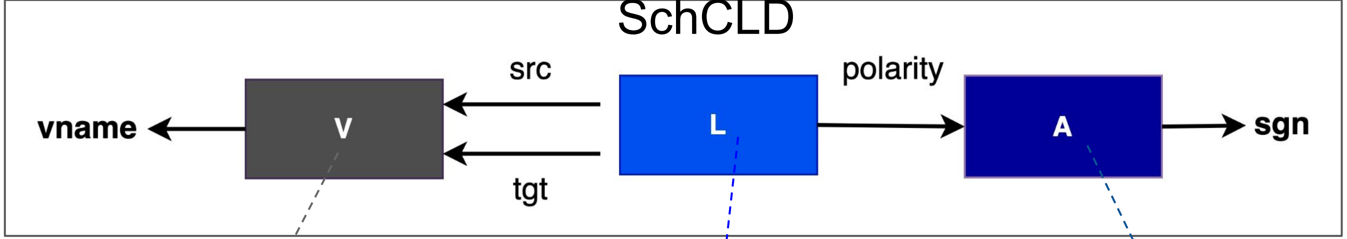
Schema of Causal Loop Diagrams (denoted as SchCLD)



CLD of smoking model, an instance of the schema, which is also a functor map: SchCLD → FinSet

- The category X of (closed) Causal Loop Diagrams is a functor category: FinSet^{SchCLD}





F(V) F(L) F(A)

Functor F Encodes as

V	vname
1	NicotineAddiction
2	Smoking
3	Health
4	CommittmentToCessation

src tgt

L	src	tgt	polarity
1	1	2	1
2	2	1	1
3	2	3	2
4	3	4	2
5	4	2	2

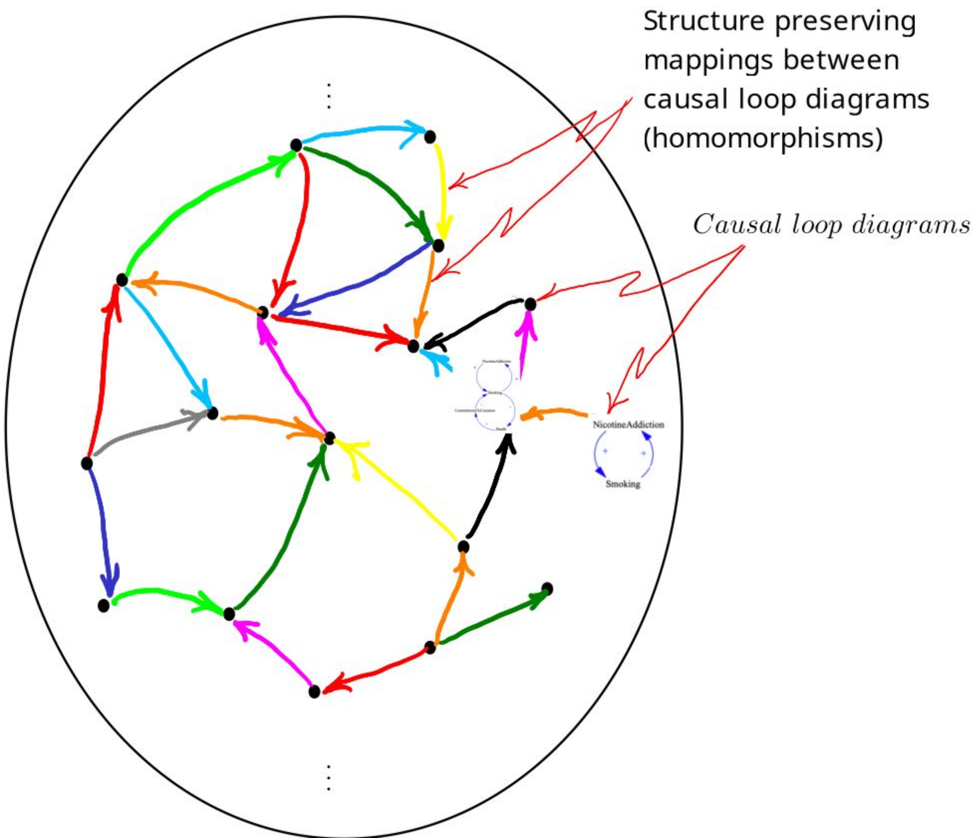
polarity

A	sgn
1	+
2	-

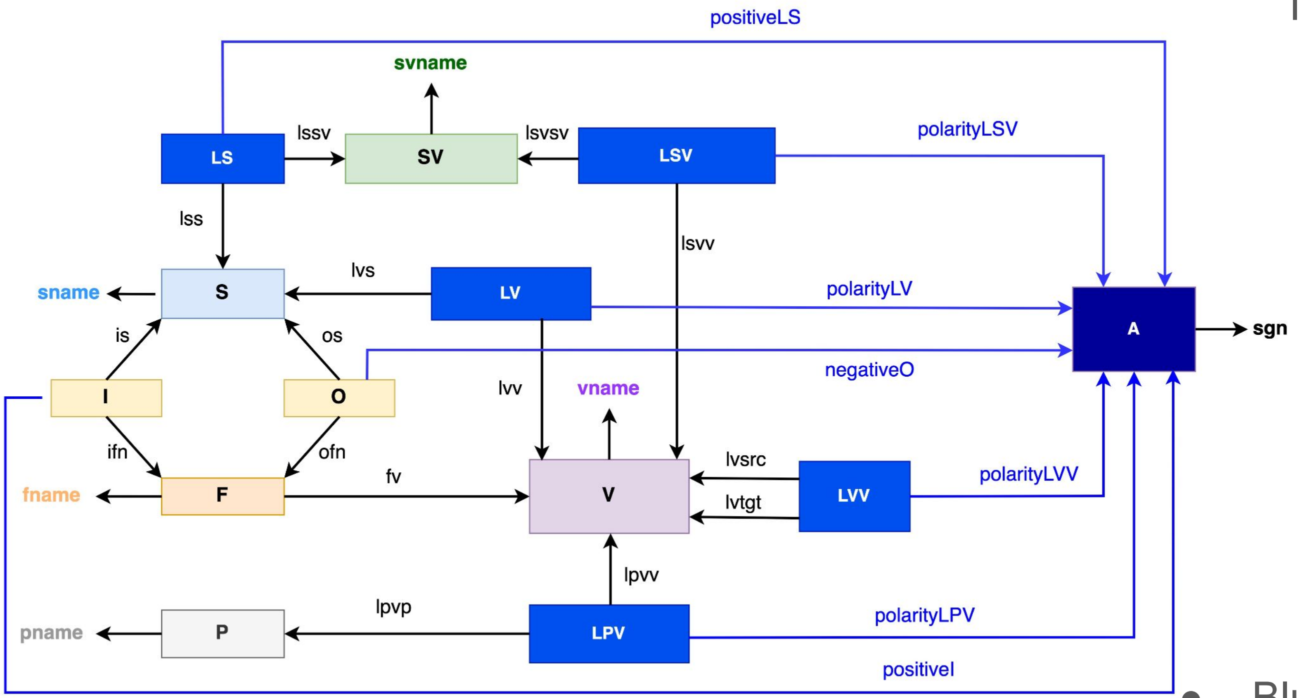
Data of the CLD are represented as a “categorical database” in software implementation:

- Each object in the schema maps to a FinSet: e.g., **Vertices**: V→{1,2,3,4}, **Links**: L→{1,2,3,4,5}, A→{1,2}
- Each object in the schema category is associated with a categorical table, and each row represents each element of the FinSet it maps to
- Each morphism (arrow) in the schema maps to a function between the FinSets to which the source and target objects of the morphism map
- Each object’s out morphism (arrow) and attribute (e.g., vname, sgn) are represented as a column of the table of the object

Category of Causal Loop Diagrams FinSet$^{\mathrm{SchCLD}}$



Category X of System Structure Diagrams: FinSet^{SchSSD}

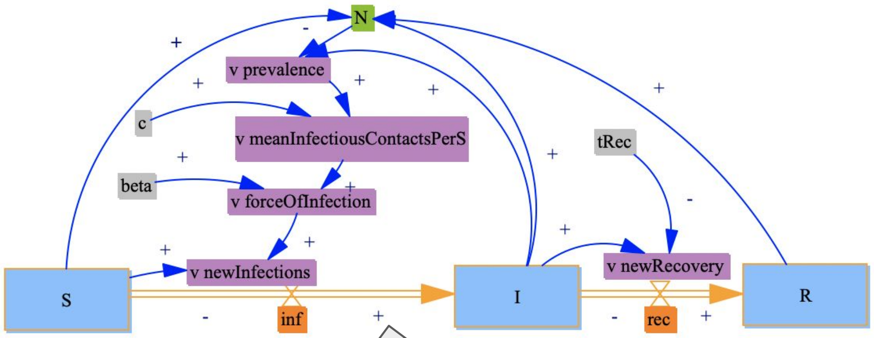


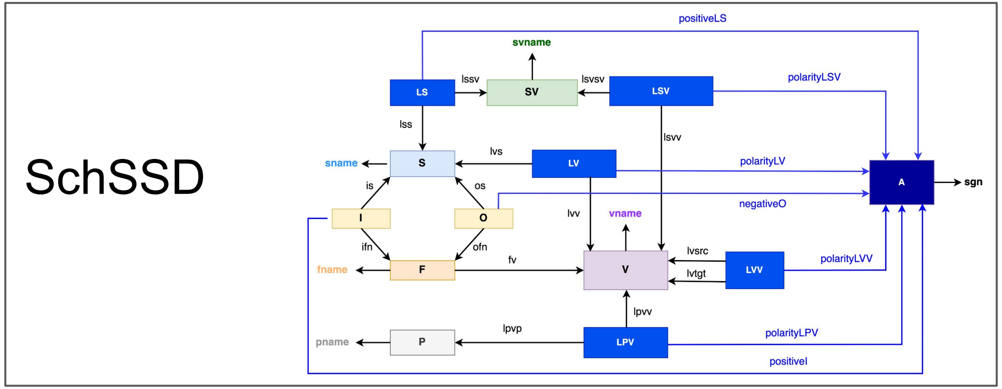
Schema of System Structure Diagrams (denoted as SchSSD)

The meaning of **objects**:

- S: Stocks
- I: Inflows
- O: Outflows
- F: Flows
- V: Auxiliary Variables
- P: Constant Parameters
- SV: Sum Auxiliary Variables
- A: Polarities ("+" and "-")
- LS, LSV, LV, LVV, LPV: Links between the two objects they connected to

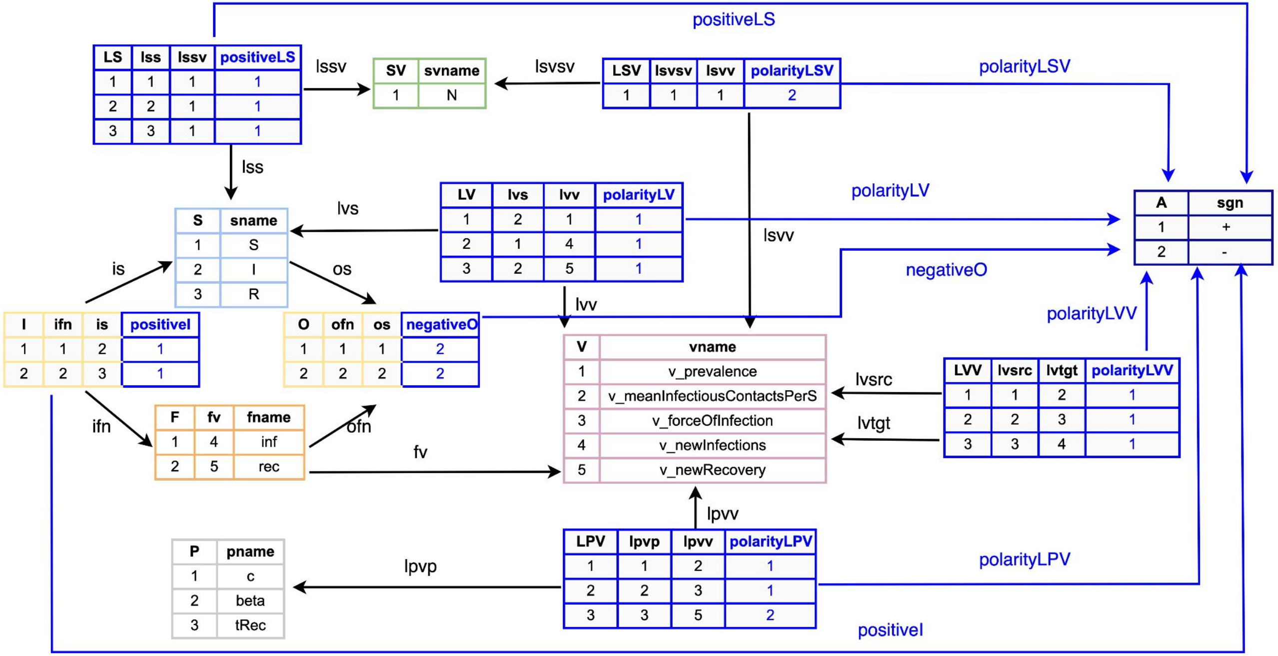
- Blue morphisms (arrows) represents the polarities
- LS and I always map to "+"
- O always map to "-"





Functor F Encodes as

Encoding System Structure Diagrams



LS	lss	lssv	positiveLS
1	1	1	1
2	2	1	1
3	3	1	1

SV	svname
1	N

LSV	lsvsv	lsvv	polarityLSV
1	1	1	2

S	sname
1	S
2	I
3	R

LV	lvs	lvv	polarityLV
1	2	1	1
2	1	4	1
3	2	5	1

A	sgn
1	+
2	-

I	ifn	is	positiveI
1	1	2	1
2	2	3	1

O	ofn	os	negativeO
1	1	1	2
2	2	2	2

V	vname
1	v_prevalence
2	v_meanInfectiousContactsPerS
3	v_forceOfInfection
4	v_newInfections
5	v_newRecovery

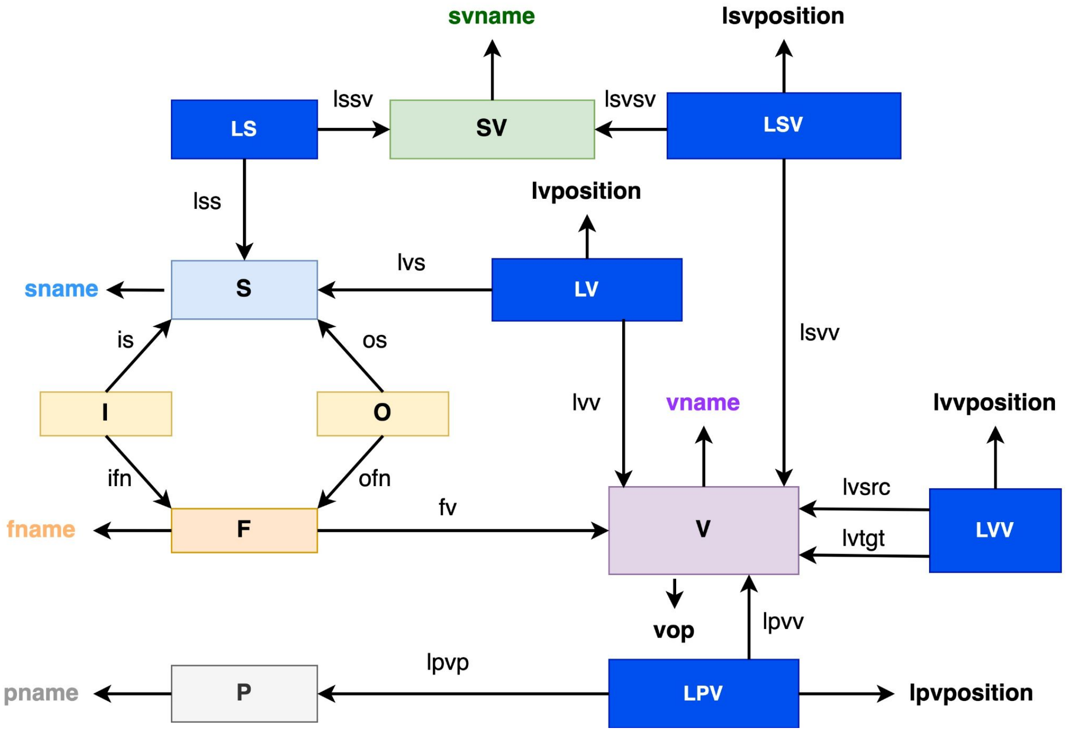
F	fv	fname
1	4	inf
2	5	rec

LVV	lvsrc	lvtgt	polarityLVV
1	1	2	1
2	2	3	1
3	3	4	1

P	pname
1	c
2	beta
3	tRec

LPV	lpvp	lpvv	polarityLPV
1	1	2	1
2	2	3	1
3	3	5	2

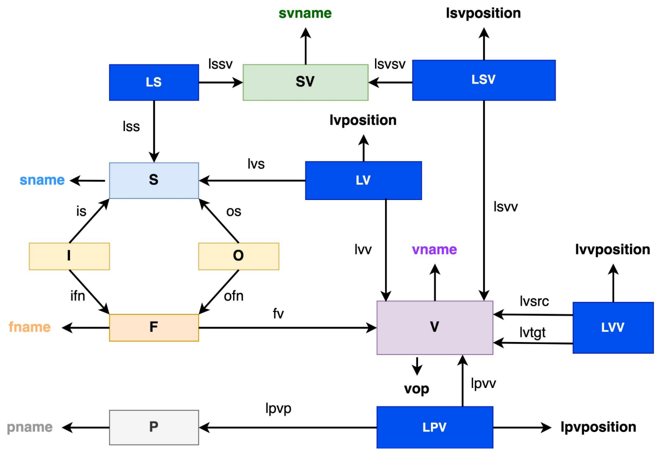
Category X of Stock & Flow Diagrams: $\mathrm{FinSet}^{\mathrm{SchSFD}}$

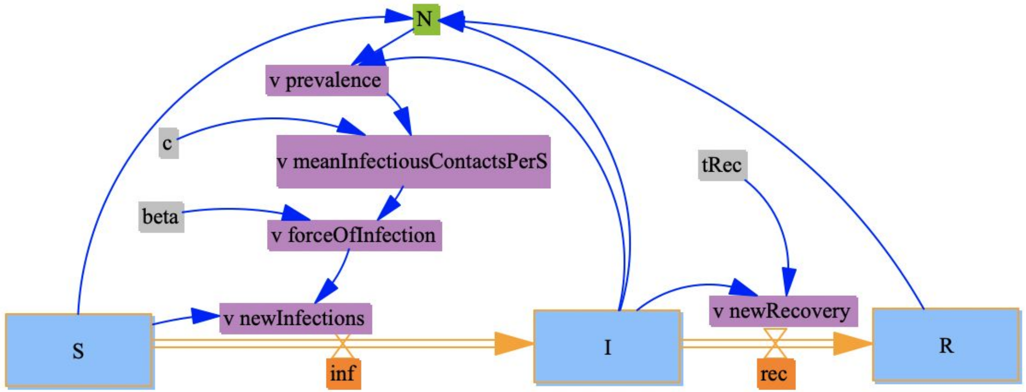


Schema of Stock & Flow Diagrams (denoted as SchSFD)

- Schema of Stock & Flow Diagrams is similar to the schema of System Structure Diagrams
- Schema of Stock & Flow Diagrams supports encoding the formulas for each auxiliary variable. Those information are represented by the attributes (“vop”, “lsvposition”, “lvposition”, “lvvposition”, “lpvposition”) of multiple objects
- Schema of Stock & Flow Diagrams does not include the information of polarities (object “A” in schema of System Structure Diagrams and all the arrows to object “A”)

Encoding Stock & Flow Diagrams





LS	lss	lssv
1	1	1
2	2	1
3	3	1

SV	svname
1	N

LSV	lsvsv	lsvv	lsvposition
1	1	1	2

S	sname
1	S
2	I
3	R

LV	lvs	lvv	lvsposition
1	2	1	1
2	1	4	1
3	2	5	1

I	ifn	is
1	1	2
2	2	3

O	ofn	os
1	1	1
2	2	2

F	fv	fname
1	4	inf
2	5	rec

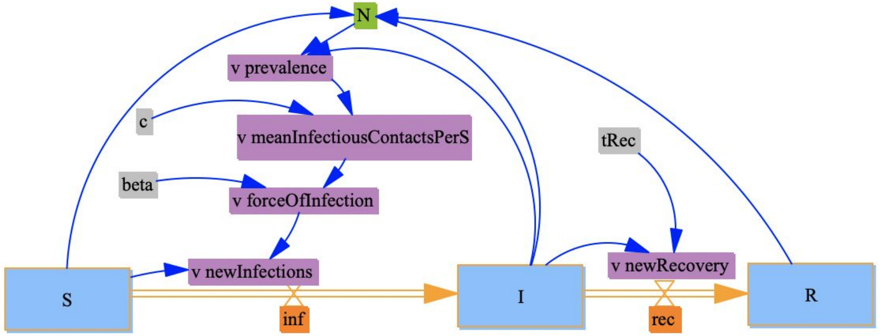
V	vname	vop
1	v_prevalence	/
2	v_meanInfectiousContactsPerS	*
3	v_forceOfInfection	*
4	v_newInfections	*
5	v_newRecovery	/

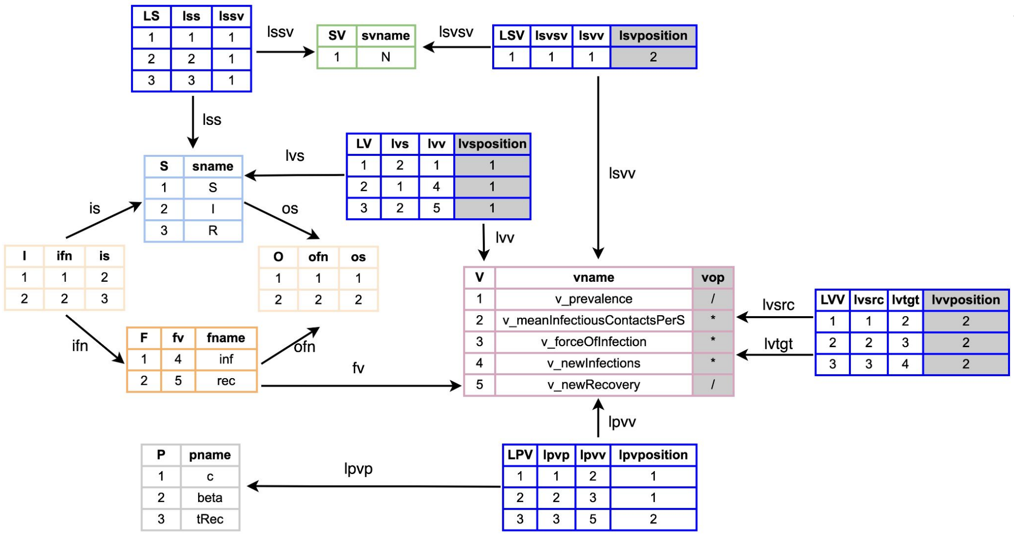
LVV	lvsrc	lvtgt	lvvposition
1	1	2	2
2	2	3	2
3	3	4	2

P	pname
1	c
2	beta
3	tRec

LPV	lpvp	lpvv	lpvposition
1	1	2	1
2	2	3	1
3	3	5	2

Algorithm for Generating the Formulas for Auxiliary Variables:





LS	lss	lssv
1	1	1
2	2	1
3	3	1

SV	svname
1	N

LSV	lsvsv	lsvv	lsvposition
1	1	1	2

S	sname
1	S
2	I
3	R

LV	lvs	lvv	lvsposition
1	2	1	1
2	1	4	1
3	2	5	1

I	ifn	is
1	1	2
2	2	3

O	ofn	os
1	1	1
2	2	2

V	vname	vop
1	v_prevalence	/
2	v_meanInfectiousContactsPerS	*
3	v_forceOfInfection	*
4	v_newInfections	*
5	v_newRecovery	/

LVV	lvsrc	lvtgt	lvvposition
1	1	2	2
2	2	3	2
3	3	4	2

F	fv	fname
1	4	inf
2	5	rec

P	pname
1	c
2	beta
3	tRec

LPV	lpvp	lpvv	lpvposition
1	1	2	1
2	2	3	1
3	3	5	2

The formulas for “Sum Auxiliary Variables (SV)” are the sum of all stocks to which they are linked. e.g., N=S+I+R. The formulas of “Auxiliary Variables (V)” are:

- The operators are represented by the attributes of “vop”. e.g., the operator of auxiliary variable “v_prevalence” is “/”
- The arguments are the source of any link among (LSV, LV, LVV, LPV) whose target are this auxiliary variable. e.g., the auxiliary variable “v_prevalence” has two arguments:
  - $lsvsv(lsvv^{-1}(v\_prevalence))=N$, with the position in the formula is “lsvposition=2”, which indicates the “N” is located as the denominator of the operator of “/”
  - $lvs(lvv^{-1}(v\_prevalence))=I$, with the position in the formula is “lvsposition=1”, which indicates the “I” is located as the numerator of operator “/”
  - Thus, the formula of “v_prevalence” is “I/N”

Symmetric Monoidal Double Category Constructed via Structured Cospan:

Let A be a category with finite coproducts, X a category with finite colimits. And L: A→X a functor preserving finite coproducts. Then, there is a symmetric monoidal double category ${}_L\mathrm{Csp}(X)$, where:

- an object is an object of A
- A vertical 1-morphism is a morphism of A
- A horizontal 1-cell is a structured cospan: $L(a) \xrightarrow{i} x \xleftarrow{o} L(b)$
- A 2-morphism is a commutative diagram:

$$\begin{tikzcd} L(a) \arrow[r, "i"] \arrow[d, "L(f)"'] & x \arrow[d, "h"] & L(b) \arrow[l, "o"'] \arrow[d, "L(g)"] \\ L(a') \arrow[r, "i'"'] & x' & L(b') \arrow[l, "o'"] \end{tikzcd}$$

Categories of Diagrams and Composing Interfaces

- Category A and X are
  - functor categories (also known as Co-presheaves, C-FinSets)
  - mapping from a small category (a schema representing the structure of
    - Diagrams
      - Stock & Flow Diagrams (X)
      - System Structure Diagrams (X)
      - Causal Loop Diagrams (X)
    - Interfaces for composition (A))

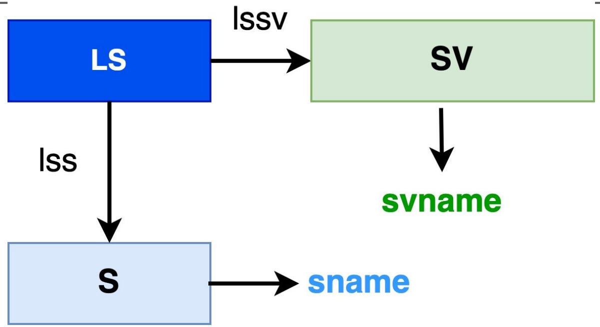
  to the category of FinSet

x (in category X) describes our system

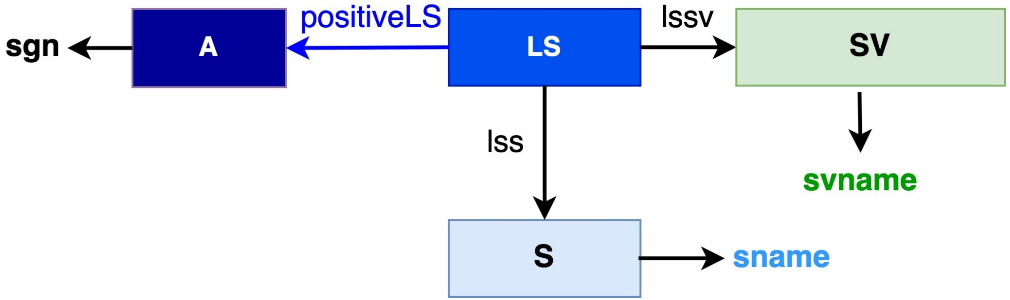
$$
\begin{tikzcd}
 & x & \\
L(a) \arrow[ur] & & L(b) \arrow[ul]
\end{tikzcd}
$$

a & b are **interfaces** (in category A)

Categories of Composing-Interfaces (Category A) – a Substructure of Category X For Each Type of Diagram

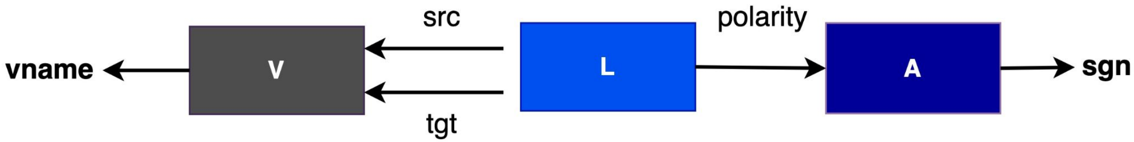


Schema of composing interface for Stock & Flow Diagrams



Schema of composing interface for System Structure Diagrams

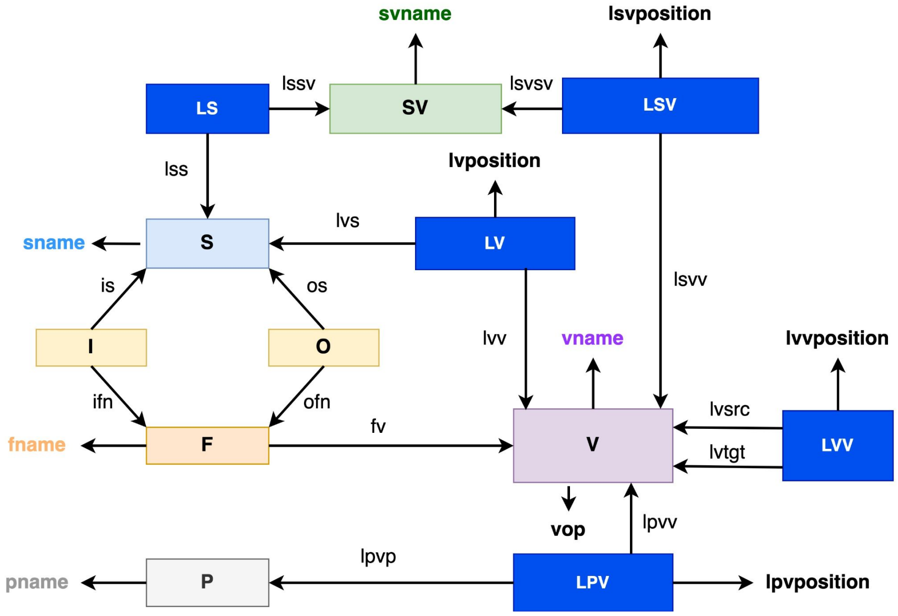
For Causal Loop Diagrams, Category A is the same as Category X:



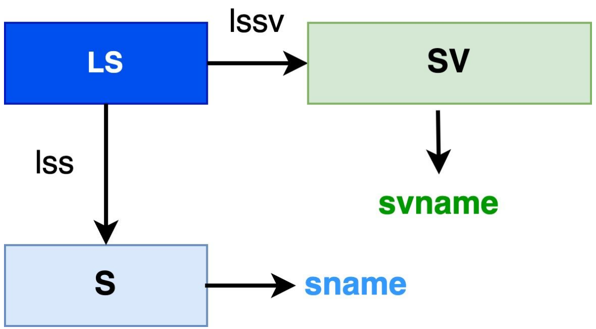
Category A for each type of diagrams is: $\mathrm{FinSet}^{\mathrm{SchInterface}}$

Example 1: Composing Open Stock & Flow Diagrams

Schemas for Diagrams & Interfaces



Schema of Stock and Flow Diagram (SchSFD)



Schema of interface (SchInterface)

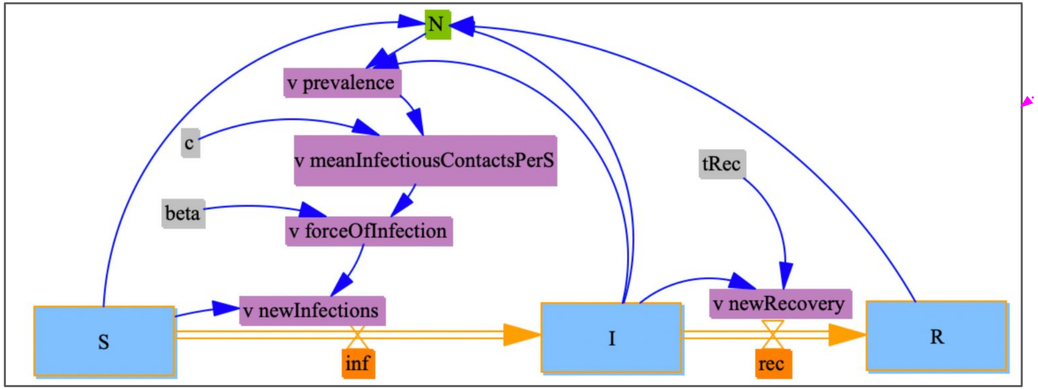
For functor L: $\mathrm{FinSet}^{\mathrm{SchInterface}} \rightarrow \mathrm{FinSet}^{\mathrm{SchSFD}}$, there is a double category ${}_{L}\mathrm{Csp}(\mathrm{FinSet}^{\mathrm{SchSFD}})$

Example 1: Composing Open Stock & Flow Diagrams

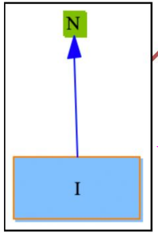
An Open Stock & Flow Diagram

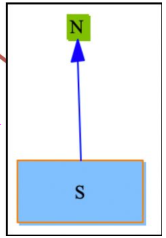
A structured cospan (proarrow, horizontal 1-cell) a→b is a diagram in category X of this form:

$$L(a) \xrightarrow{i} x \xleftarrow{o} L(b)$$



Instance of Schema of Stock and Flow Diagram (SchSFD)





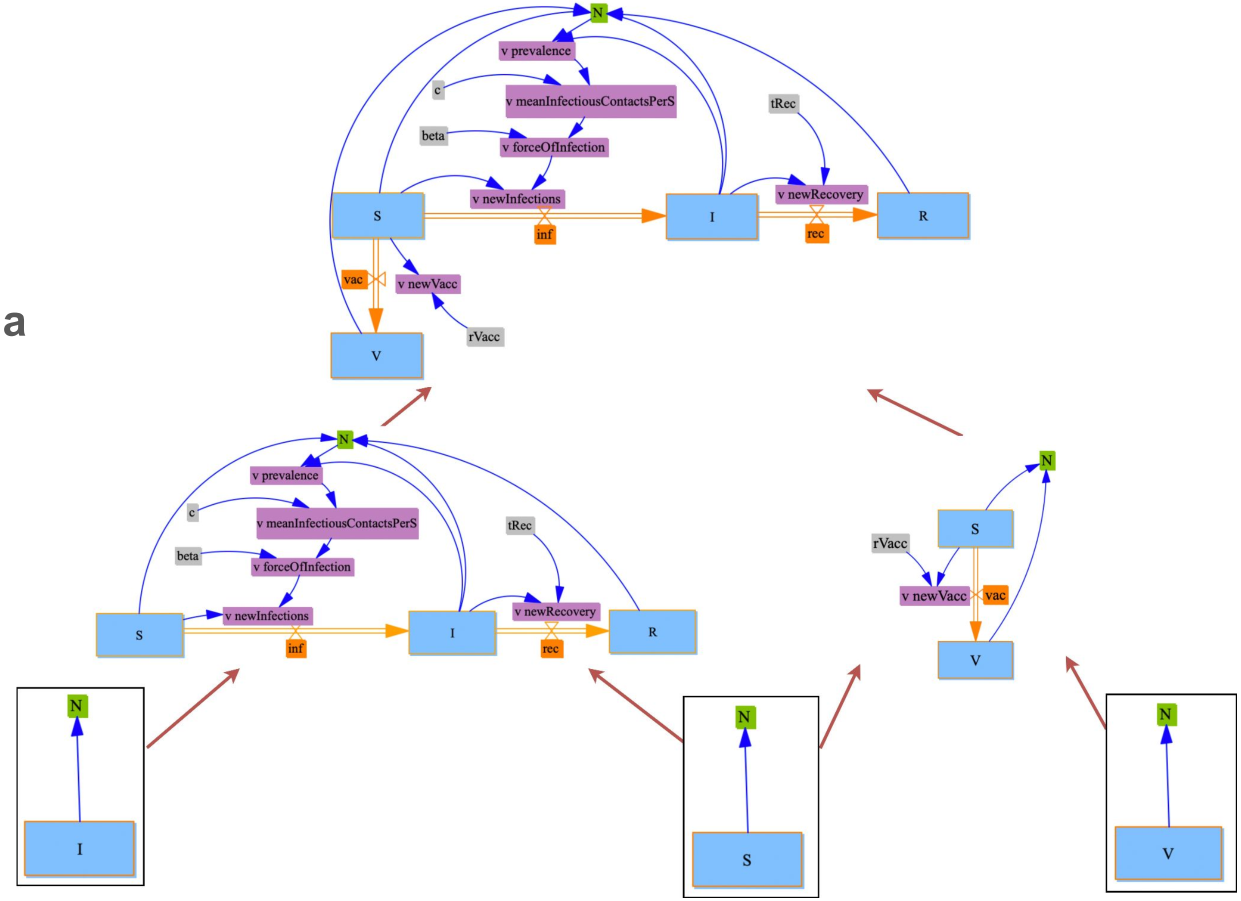
Instances of schema of interface (SchInterface)

Example 1: Composing Open Stock & Flow Diagrams

Composition of Open Stock & Flow Diagrams

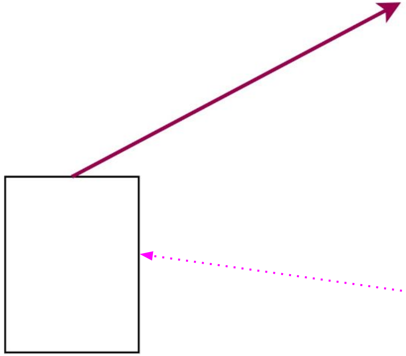
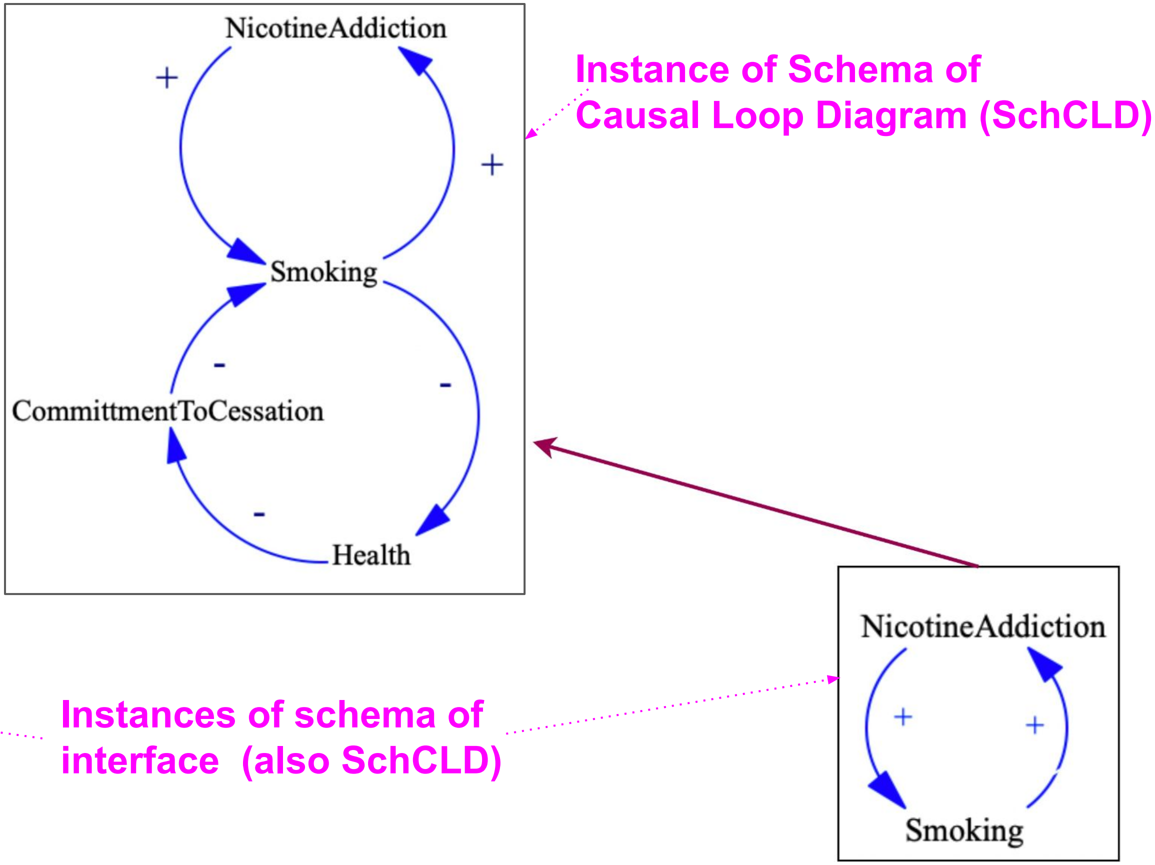
$$\begin{tikzcd}
 & & x +_{L(b)} y & & \\
 & x \arrow[ur, "j_x"] & & y \arrow[ul, "j_y"'] & \\
L(a) \arrow[ur, "i_1"] & & L(b) \arrow[ul, "o_1"'] \arrow[ur, "i_2"] & & L(c) \arrow[ul, "o_2"']
\end{tikzcd}$$

Composition is Achieved via Pushout



Example 2: Composing Open Causal Loop Diagrams

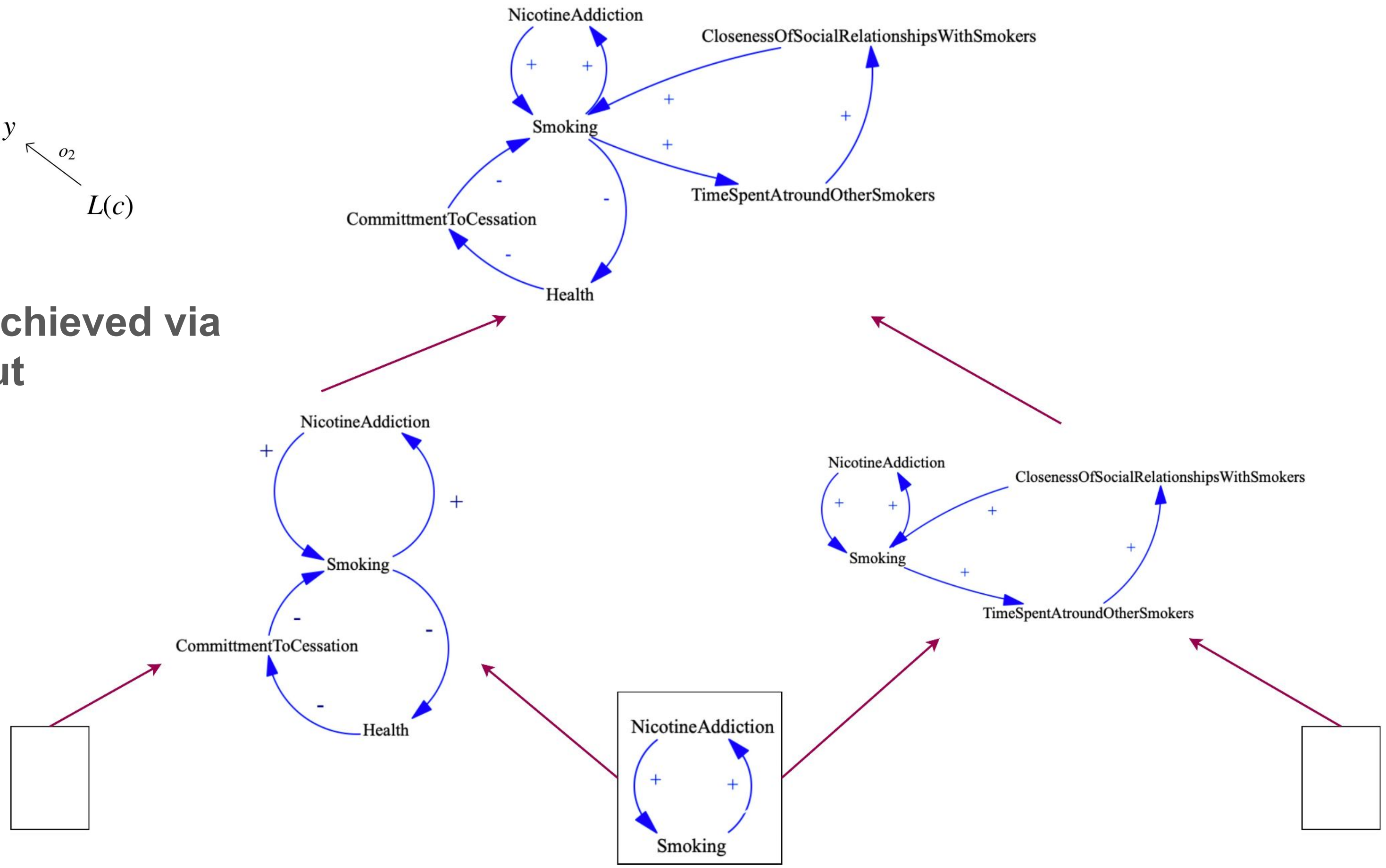
Simply, a open causal loop diagram is a cospan structure in Category X of Causal Loop Diagrams:

Example 2: Composing Open Causal Loop Diagrams

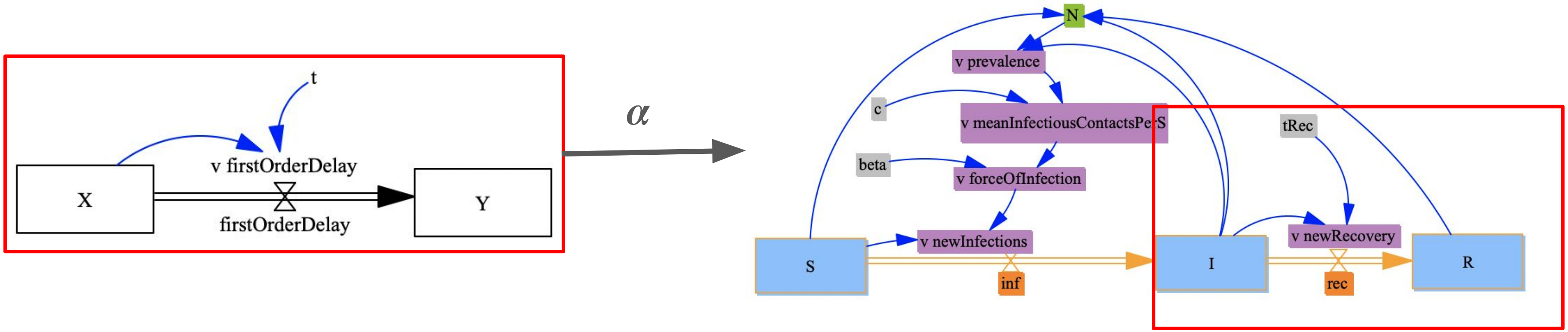
$$\begin{tikzcd}
 & & x +_{L(b)} y & & \\
 & x \arrow[ur, "j_x"] & & y \arrow[ul, "j_y"'] & \\
L(a) \arrow[ur, "i_1"] & & L(b) \arrow[ul, "o_1"'] \arrow[ur, "i_2"] & & L(c) \arrow[ul, "o_2"']
\end{tikzcd}$$

Composition is Achieved via Pushout



Homomorphisms: Identifying Patterns

Example: Identifying the Pattern of First Order Delay



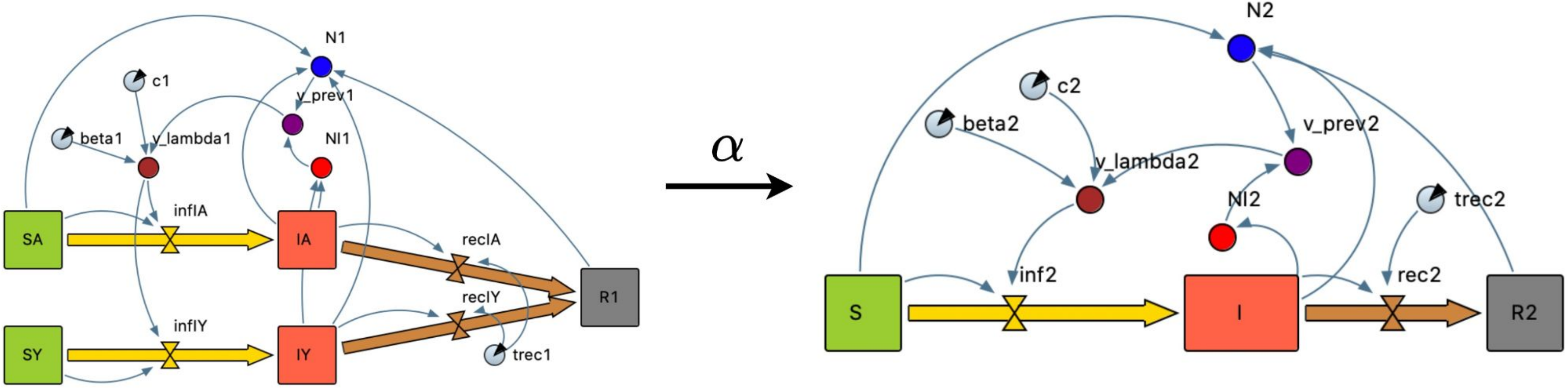
α(x) = I
α(Y) = R
α(t) = tRec

α(firstOrderDelay) = rec

α(v_firstOrderDelay) = v_newRecovery

Homomorphisms: Typed Diagrams

Homomorphism (Structure Preserving Mapping) of Stock & Flow Diagrams



$\alpha(SA) = S$

$\alpha(SY) = S$

$\alpha(IA) = I$

$\alpha(IY) = I$

$\alpha(R1) = R2$

Stocks

$\alpha(infIA) = inf2$

$\alpha(infIY) = inf2$

$\alpha(recIA) = rec2$

$\alpha(recIY) = rec2$

Flows

$\alpha(N1) = N2$

$\alpha(NI1) = NI2$

Sum Auxiliary Variables

$\alpha(c1) = c2$

$\alpha(beta1) = beta2$

$\alpha(trec1) = trec2$

Parameters

$\alpha(v\_prev1) = v\_prev2$

$\alpha(v\_lambda1) = v\_lambda2$

Auxiliary Variables

Stratification: Typed Diagrams

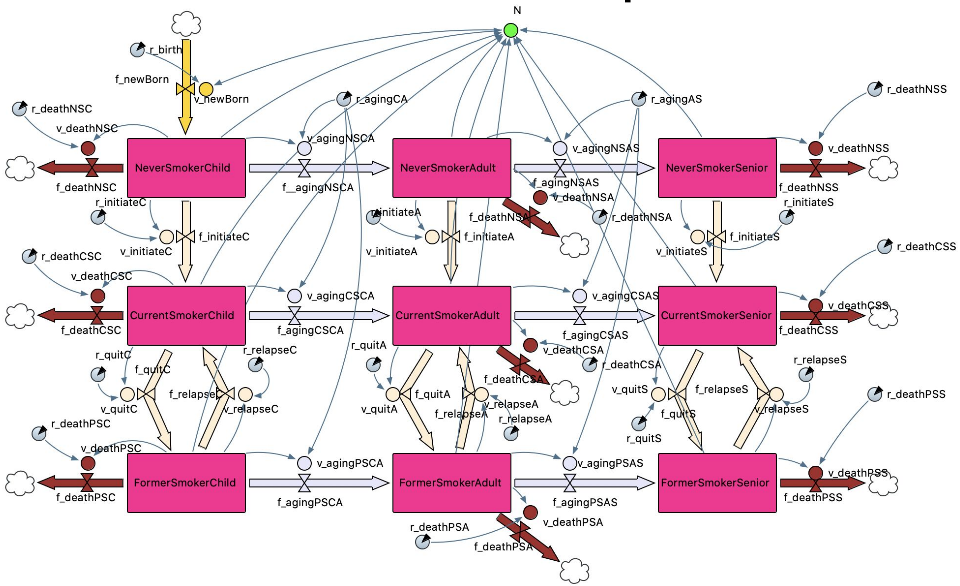
Mathematically, the stratified diagrams are taking “pullback” (or “limit”) of diagrams in the category X. And the pullback square is:

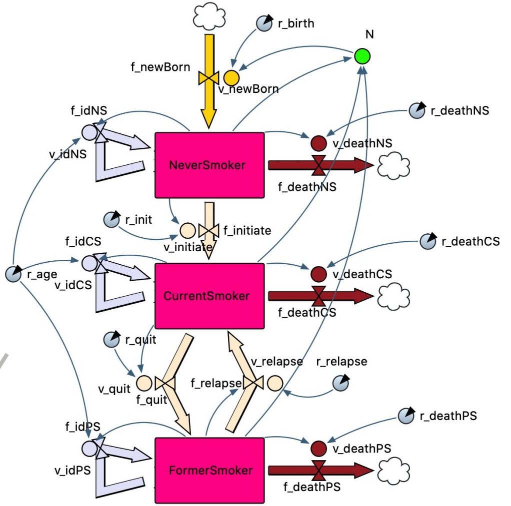
$$
\begin{tikzcd}
S_{stratified} \arrow[r, "p_1"] \arrow[d, "p_2"'] \arrow[dr, phantom, "\lrcorner", very near start] \arrow[dr, phantom, "\checkmark"] & S_{aggregate} \arrow[d, "t_{aggregate}"] \\
S_{strata} \arrow[r, "t_{strata}"'] & S_{type}
\end{tikzcd}
$$

Structure-preserving mapping (Homomorphism)

Type Diagram

Stratification: An Example of Stratified Stock & Flow Diagrams





example

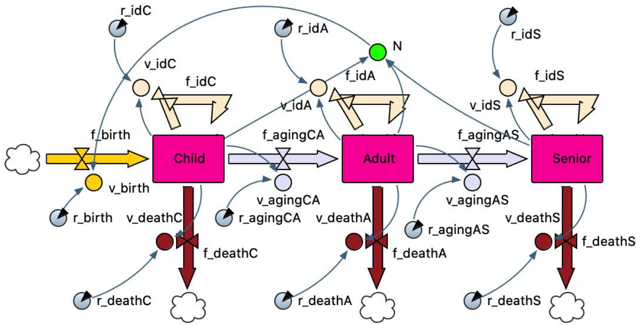
example

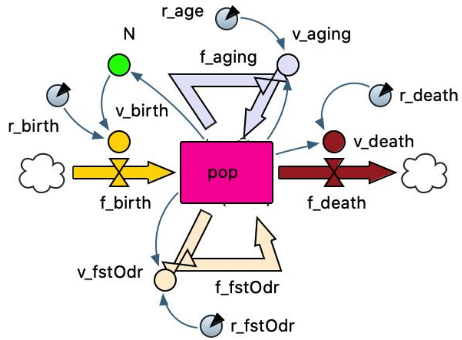
$$
\begin{tikzcd}
S_{stratified} \arrow[r, "p_1"] \arrow[d, "p_2"'] \arrow[dr, phantom, "\checkmark"] & S_{aggregate} \arrow[d, "t_{aggregate}"] \\
S_{strata} \arrow[r, "t_{strata}"'] & S_{type}
\end{tikzcd}
$$

Pullback Square

example

example





Composable Maps among Three Diagram Types

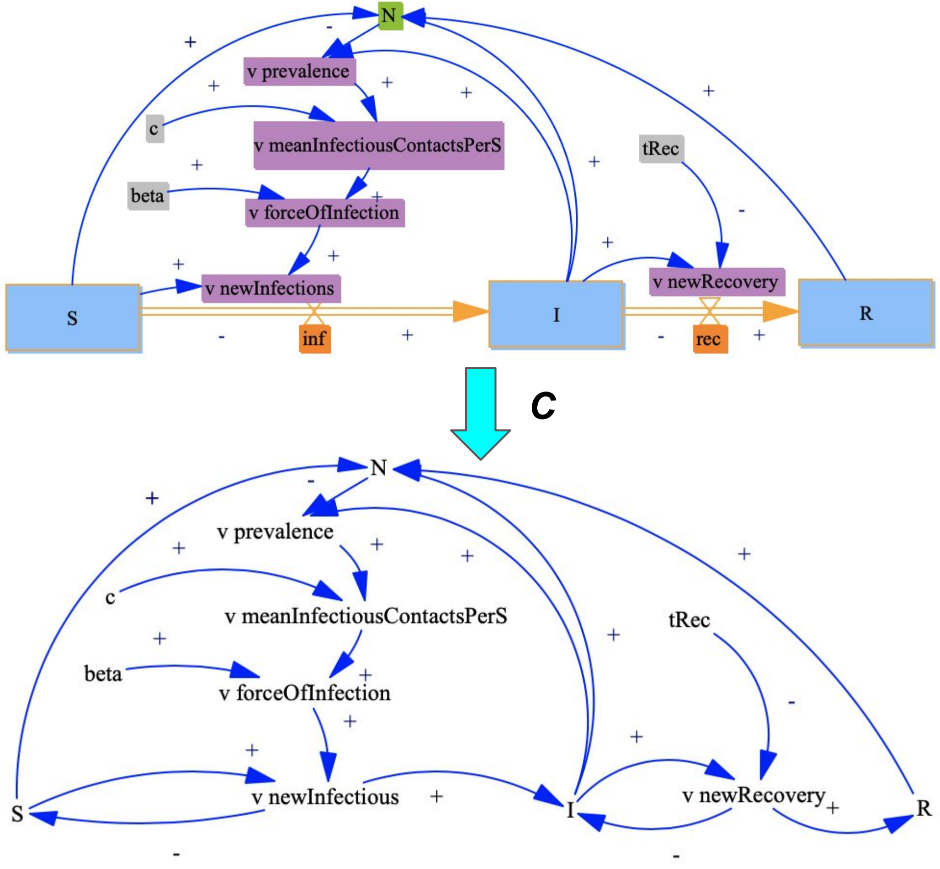
$$
\begin{tikzcd}
StockFlow \arrow[rr, "B"] \arrow[dr, "A"'] & & CausalLoop \\
& SystemStructure \arrow[ur, "C"'] &
\end{tikzcd}
$$

Capturing the Functor Maps between Diagram Types: Causal Loop Diagrams from System Structure Diagrams

The functor C: $\mathrm{FinSet}^{\mathrm{SSD}} \rightarrow \mathrm{FinSet}^{\mathrm{CLD}}$ is specified by the functor CLD → the category of duc-queries on SSD using Data Migration:

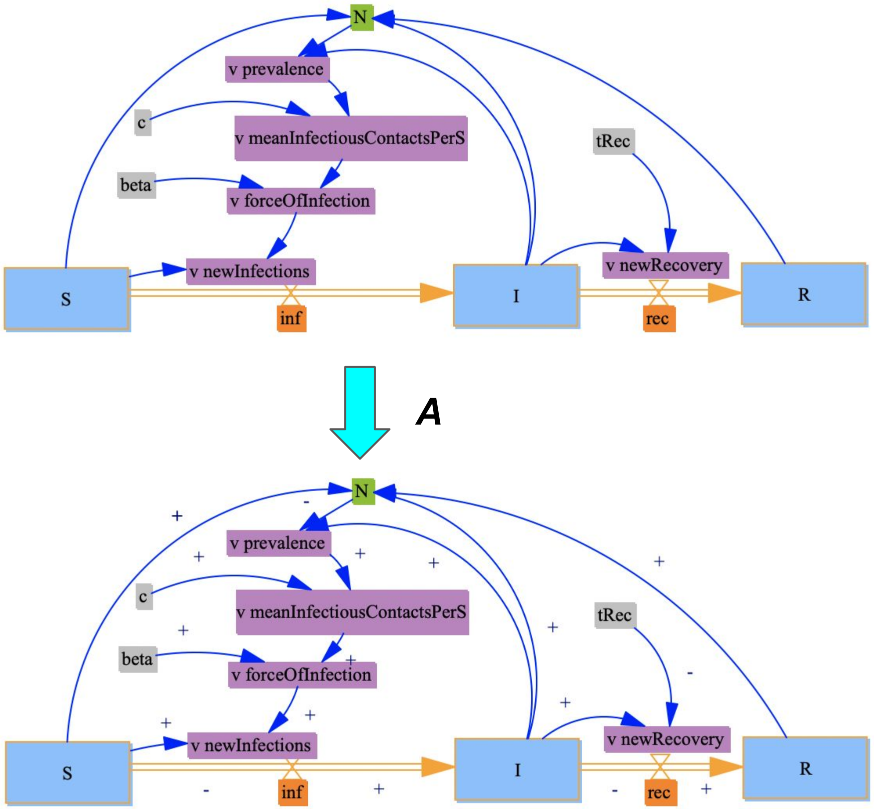
SchCLD → the category of duc-queries on SchSSD:

V → S+SV+V+P
L → LV+LS+LSV+LPV+LVV+I+O
A → A
src → [lvs, lss, lsvsv, lpvp, lvsrc, ifn;fv, ofn;fv]
tgt → [lvv, lssv, lsvv, lpvv, lvtgt, is, os]
polarity → [polarityLV, positiveLS, polarityLSV, polarityLPV, polarityLVV, positiveI, negativeO]



Capturing the Interconnections between Diagram Types: System Structure Diagrams from Stock & Flow Diagrams

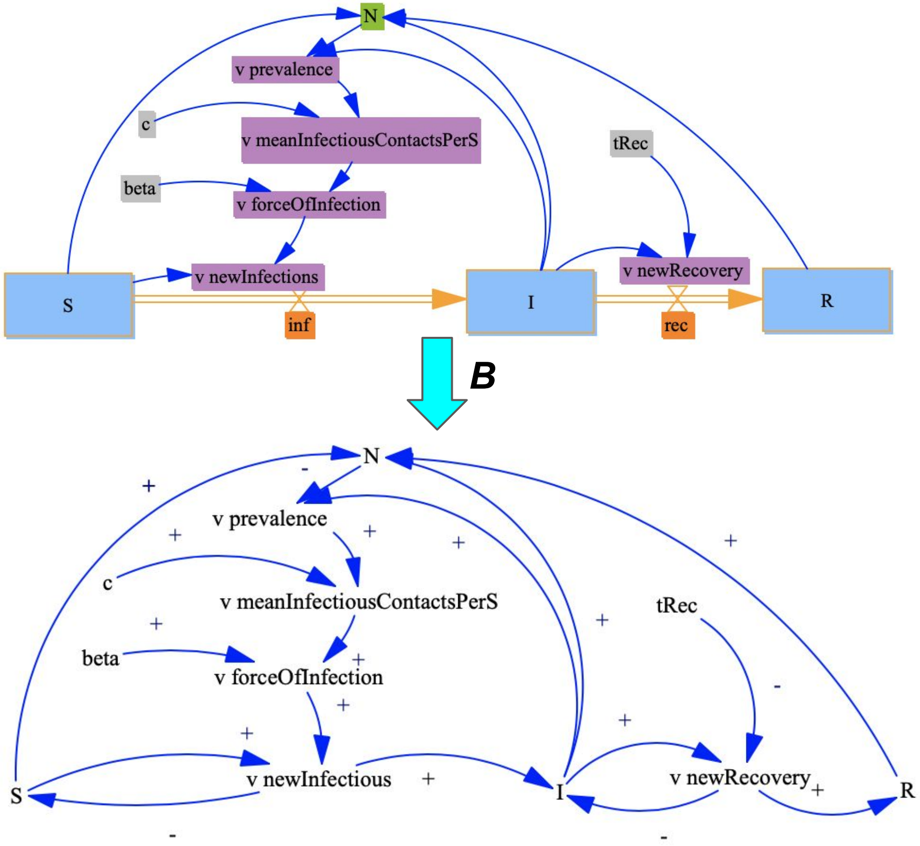
- Compared to the schema of Stock & Flow diagrams, that for System Structure diagrams does not include *attributes* of:
  - “Operators” of each dynamical variables.
  - Attributes of “positions” of each argument of the expressions of dynamical variables.
- For morphisms of polarities: Polarities of links
  - from stocks to sum-dynamic variables are always “+”
  - from outflows to upstream stocks are always “-”
  - from inflows to downstream stocks are always “+”
  - from stocks, sum-dynamic variables, dynamic variables and constant parameters to dynamic variables are calculated depending on the “operators” of the target dynamic variables and “positions” of those arguments. e.g., if the target dynamic variable’ operator is “minus”, and the source argument’s position is “2” and assume the argument is always >0, then the polarity is “-”.



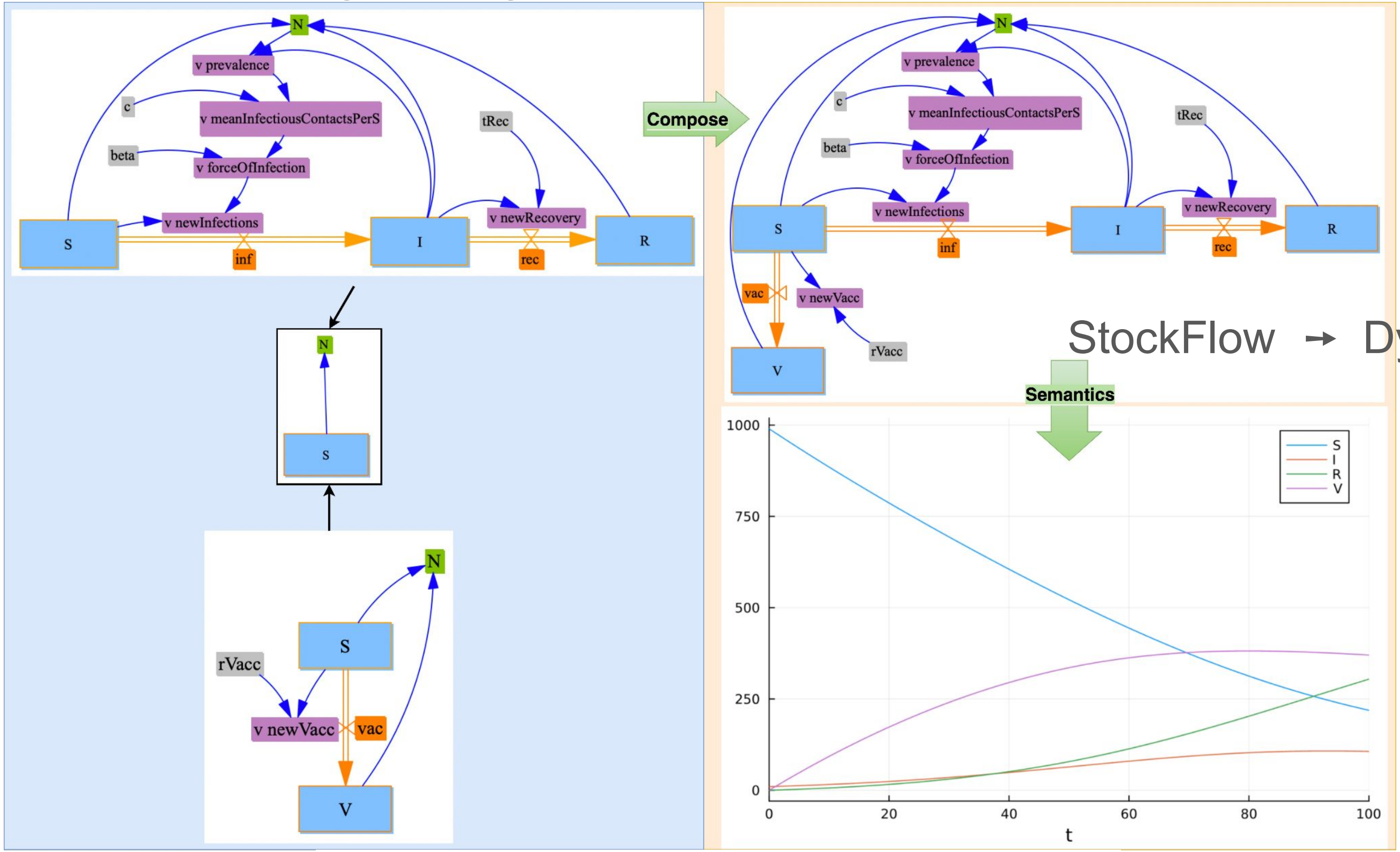
Capturing the Interconnections between Diagram Types: Causal Loop Diagrams from Stock & Flow Diagrams

The map from Stock & Flow Diagrams to Causal Loop Diagrams can be simply captured by the combination of the map from Stock & Flow Diagrams to System Structure Diagrams and the map from System Structure Diagrams to Causal Loop Diagrams.

$$\begin{tikzcd} StockFlow \arrow[rr, "B"] \arrow[dr, "A"'] & & CausalLoop \\ & SystemStructure \arrow[ur, "C"'] & \end{tikzcd}$$



Mapping Diagrams to Semantic Domains



StockFlow → Dynam

$$d(\text{Stock}) / dt = \text{sum}(\varphi_{\text{Inflows}}) - \text{sum}(\varphi_{\text{Outflows}})$$

Examples of Possible Semantics

- Stochastic discrete transitions [Implemented]
- Simulation
- Calibration
- Loop gain analysis
- Eigenvalue elasticity analysis
- Unit diagnostics
- [With augmented data, Stochastic continuous flows, Kalman filtering, Particle filtering & Particle MCMC via Markov categories]

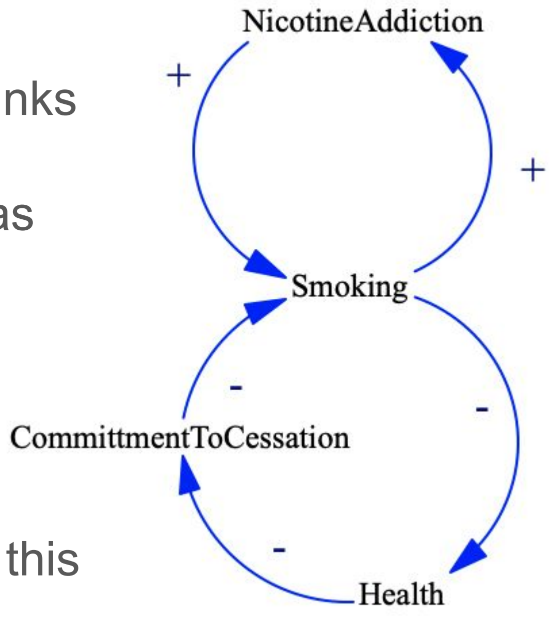
Further Needs

- Capturing compositionality for links, pathways & loops polarities
- Identifying pathways & loops with polarities

Category of Signed Category representing CLDs supporting Identifying both Links & Loop Polarities

More advanced categorical structure (double theories) allows representing CLDs in a way that supports both links & loop polarities and path composition:

- The previous categorical structure we treat CLDs as signed graphs
- Now, we treat each CLD as a **signed category**:
    - Each variable is an object
    - Each link is a morphism
    - Each link companions with a sign “+” or “-”
    - Composing paths is composing morphisms in this signed category, and the composed polarity is calculated by the multiplication of the polarities of the morphisms being composed

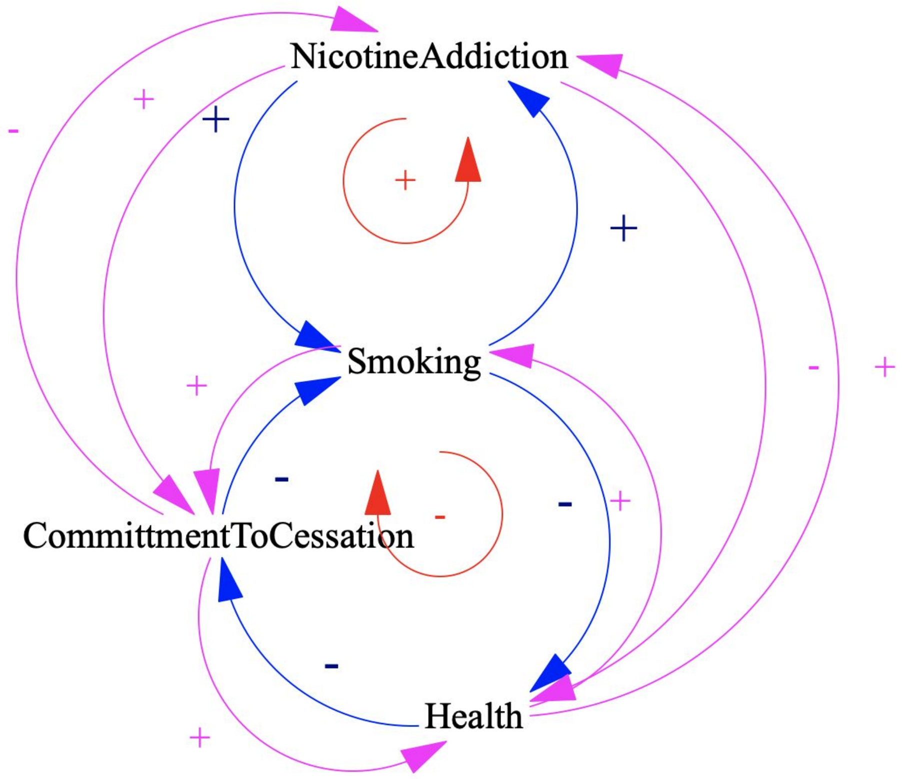


This is a category

Category of Signed Category representing CLDs supporting Identifying both Links & Loop Polarities

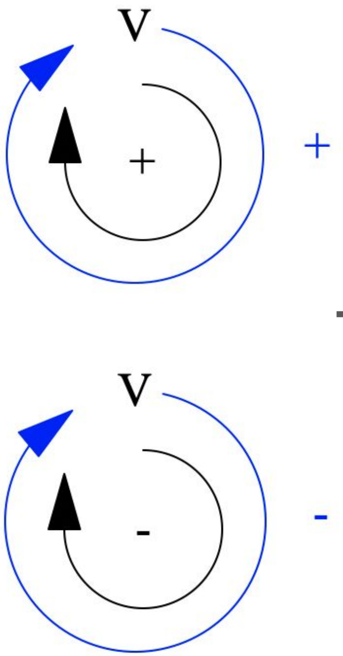
An example:

- Blue links are the basic ("generator") morphisms representing this signed category by signed graph
- Magenta links are the composed paths generated by composing the morphisms with multiplying the polarities
- We then are able to determine the loop polarities



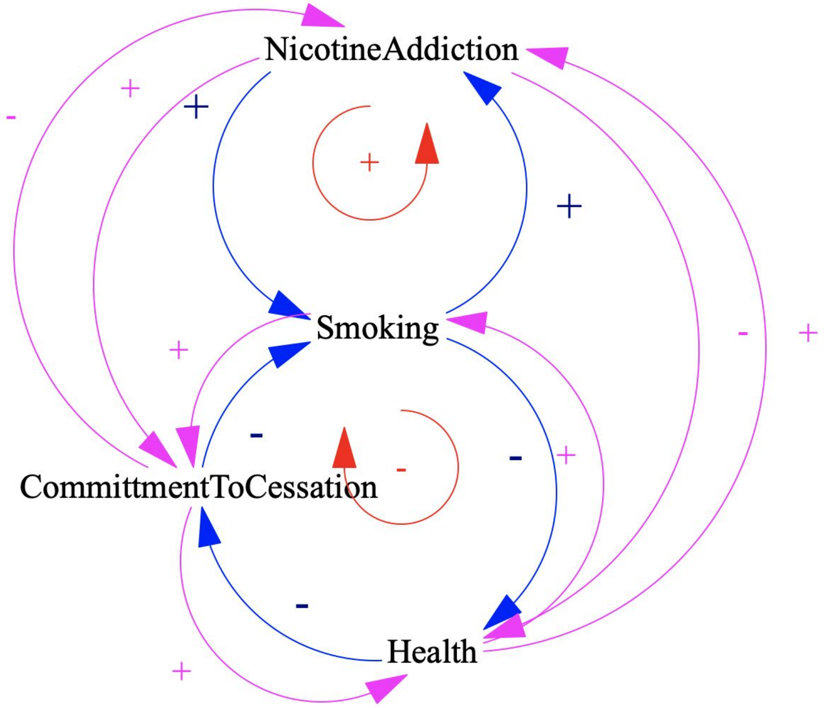
Identifying Loop Polarities

Two patterns



Identify by functor mapping

Each Instance is Matched by 1 Functor



Priorities for Coming Work: SD Theory

- Incorporation of dimensional/unit annotations
- Interoperability support: Categories of open dynamical systems
- Semantics
  - Loop gain
  - Eigenvalue elasticity analysis
  - Stochastic discrete transitions
  - Unit diagnostics
  - [With augmented information] Stochastic continuous
  - [With augmented information] Particle filtering & Particle MCMC via Markov categories
- Summaries of temporal behaviour over differing scales via temporal sheaves
- Support for separate representation of model logic & presentation
- Topos-based query language for reasoning about behaviour

Modeling Process Support

- Gaps: Lack of mathematical structure of processes
  - Marshalling team contributions & needs
  - Tracing provenance information
  - Capturing team interactions around models & modeling
  - Versioning evolution
- Categorial methods allow for capturing aspects of the modeling process and evolution distinct from model

Parallel Priority: Compositional Agent-Based & Hybrid Modeling

- Goal: A categorical mathematical foundation & software implementation (AlgebraicABMs) for a broad class of agent-based & hybrid models
  - Declarative rules (Statecharts, Rewrite rules, stock & flow)
  - Modularity
  - Metalinguistic abstraction
  - Relational/contextual situations recognized as first-class entities
  - Foundational support for hybrid ABM & stock-flow modeling
- Joint work with John Baez (UC Riverside & U Edinburgh & Topos Institute), Kris Brown & Evan Patterson (Topos), William Waites (U Southampton)
- Key locus of this work: ICMS (U Edinburgh) in May-June 2024 & 2025

Take-Home Messages

- ACT can explicate & help realize the potential of the mathematics underlying the spectrum of System Dynamics diagrams
- ACT can rigorously relate diagrams to others of the same and different types
- ACT can empowers teams via transparency, modularity, composition, abstraction, interconnections between diagram types, easier exploration of model structure, cleaner stratification & enhanced semantic flexibility
- ACT-based systems can support System Dynamics modeling in teams without end-user familiarity with category theory
- Power SD users can use our evolving toolsets to manipulate, analyze and map SD models in flexible manners
- Future evolution of the toolsets are expected to support deeper manipulation and analysis of model structure and behaviour, and the modeling process