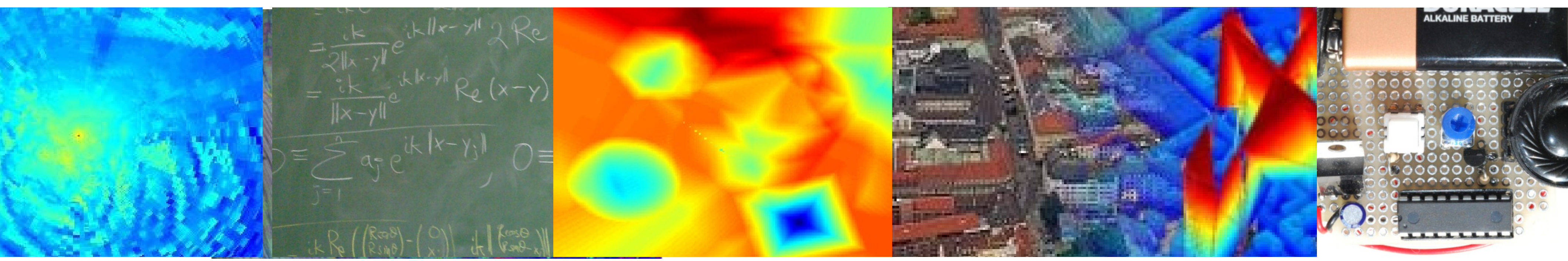
# Practical Systems Modeling *in* Categories using Sheaves




Michael Robinson

# Acknowledgments

This is joint work with a number of people in different roles:

- Collaborators:
  - Letitia Li, Cory Anderson, Denley Lam (BAE Systems)
  - Kris Ambrose, Steve Huntsman, Allyson O'Brien, Matvey Yutin (all formerly at BAE Systems)
  - Cliff Joslyn, Brenda Praggastis, Emilie Purvine (PNNL)
  - Chris Capraro, DJ Isereau (SRC)
  - Donna Dietz (American University) ← Python Colab notebooks!
- Numerous students at American University
- Funding:
  - DARPA, ONR, AFRL, PNNL, AU, among others

# Categories!

- Category theory is a good way to organize compositional models of systems
- Category theory is the "theory of mathematical analogies"
- Surprisingly, it's expressive enough to represent mathematical logic!
- But in a sense, that's too expressive… it's hard to get a handhold
- Sheaf theory is a part of category theory, and provides some useful constraints to simplify modeling

# Why sheaves?

Sheaves:

- Are the **univeral reductionist paradigm** that guide the composition of more complicated models from simpler ones
- Moderate between **different levels of abstraction** and/or domains of validity for models

And recently, they can handle noisy real-world data with practical models in software

# Sheaves are universal

- Theorem: (Lawvere, 1960s & 70s) Formal logic can be encoded in a category of sheaves
  - All nontraditional logics can be encoded as well
  - (this hasn't been realized in software; it's somehow too unwieldy...)
- Theorem: (R., 2017) Any framework that assembles local models into global ones consistently will yield sheaves

Hence,

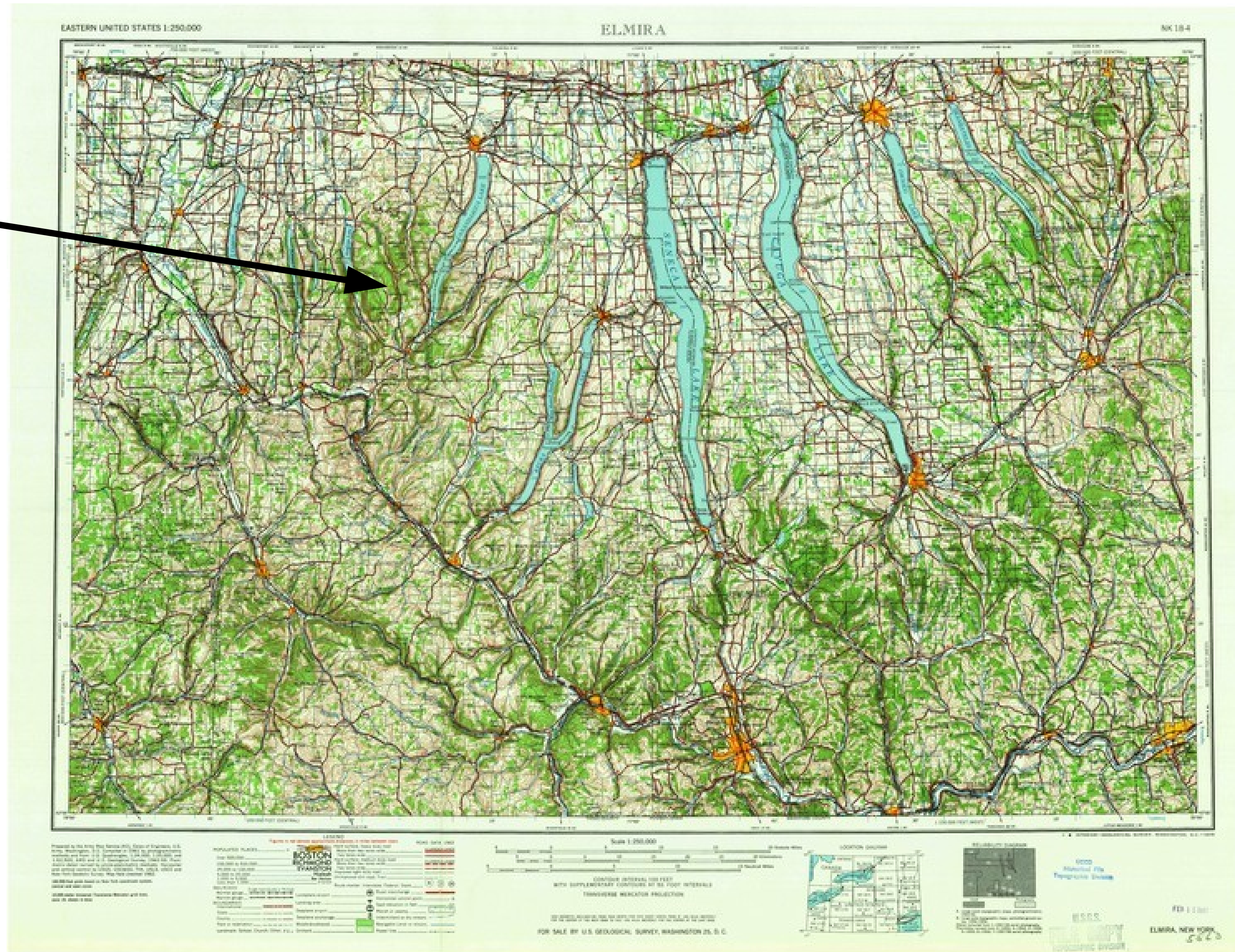
- Any systematic reductionist approach to science entails the use of sheaves, at least in part

# Sheaves guide the level of abstraction

- George Box (1978), "All models are wrong but some are useful"
- Question: What is the *domain of validity* for a model?
  - Box, again, "It is inappropriate to be concerned about safety from mice when there are tigers abroad."
- Claim: This is a *topological* notion

# What is topology?

Not this!
This is
TopoGRAPHY!




(Thanks USGS!)

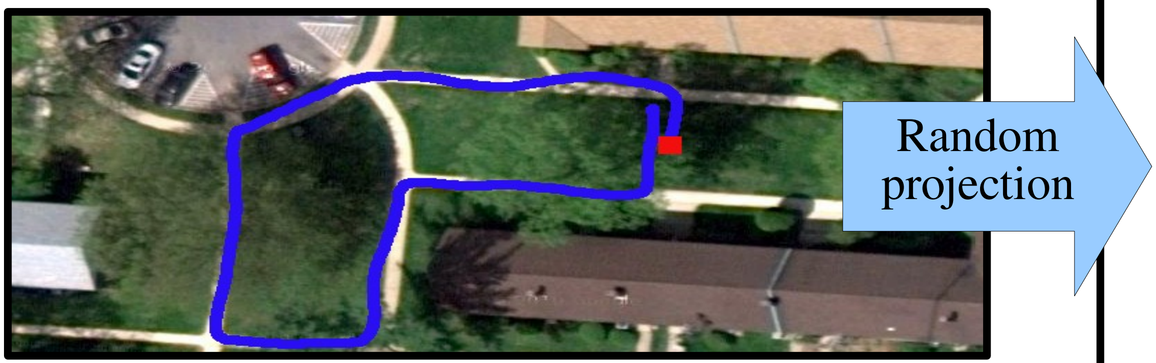
# What is topology?

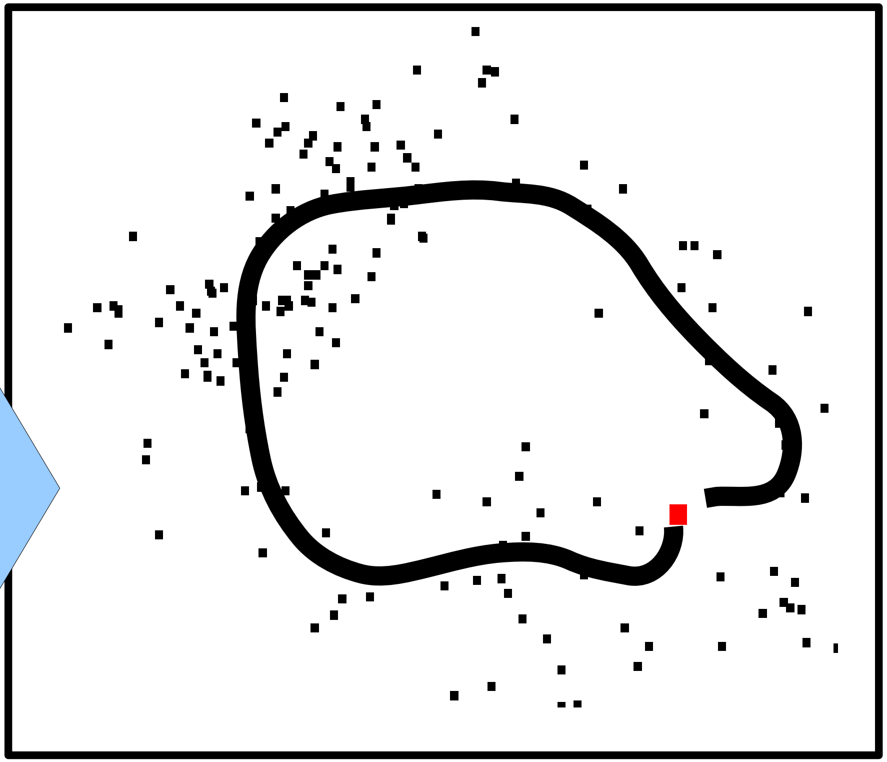
=

Topology is the study of spaces under continuous deformations

# Topology is present in data

- Cellphone used to record signal level of 802.11 access points near several apartment buildings
- Signal level (dBm) and station MAC address recorded periodically
  - Uniquely identified 52 WAPs
- Random projection to 2d



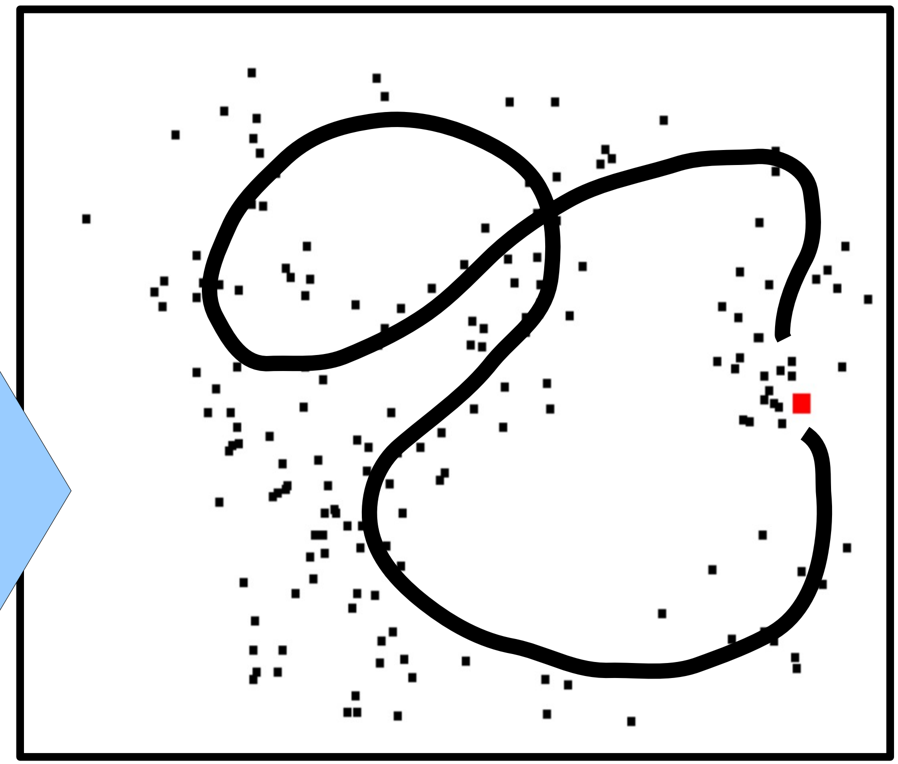



(image courtesy of Google)

Software credit: Daniel Muellner and Mikael Vejdemo-Johansson

# Topology is present in data

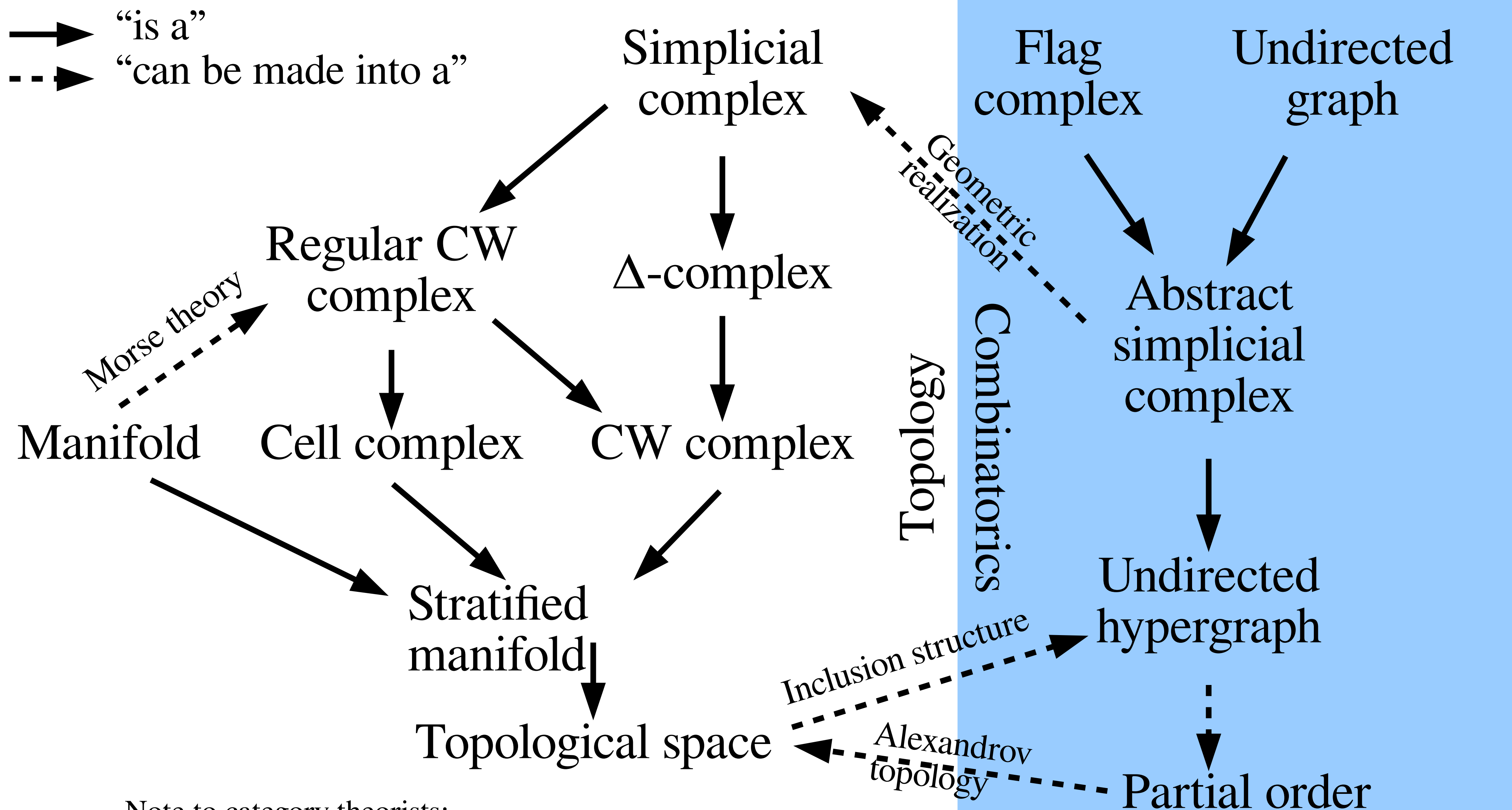
- Goal: measure environment and targets with minimal sensing and opportunistic sources
- Key theoretical guarantees proven
- First generation algorithms
  - Simulated extensively
  - Validated experimentally

(image courtesy of Google)

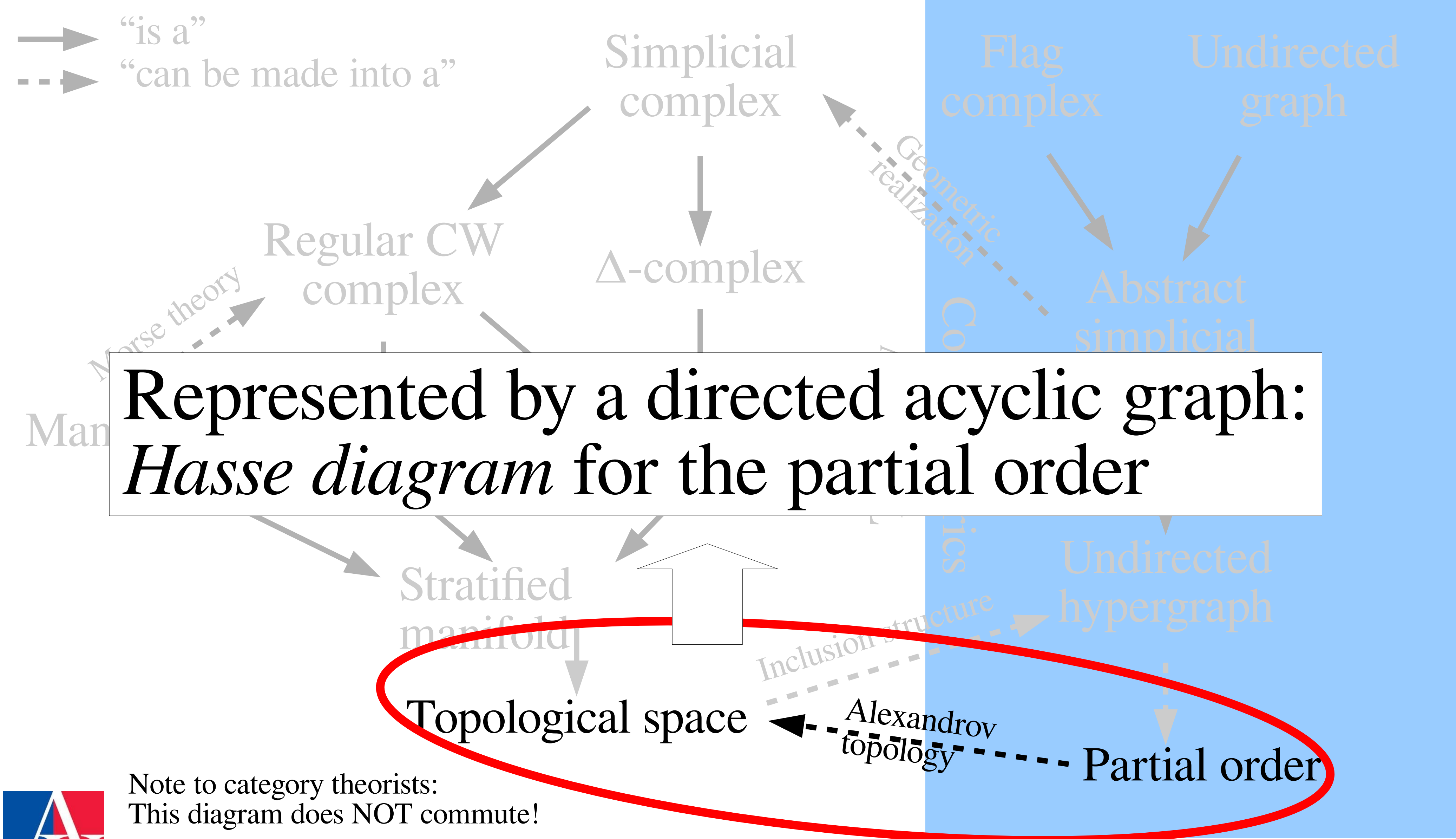
Software credit: Daniel Muellner and Mikael Vejdemo-Johansson

# There are lots of topologies...




Note to category theorists:
This diagram does NOT commute!

# In practice, we only need these...




Represented by a directed acyclic graph:
*Hasse diagram* for the partial order

Note to category theorists:
This diagram does NOT commute!

# A sheaf relates Topology → Models

- A *sheaf* is a data structure that:
  - Pairs a domain of validity with a corresponding model, and

    Graph node: *cell* — Distance metric: *stalk*
  - Explains how the model changes with the domain

    Function on each graph edge: *restriction*

We say "sheaf ON a partial order OF metric spaces"

- Formally: a *sheaf* is a functor from the partial order to the category of metric spaces & continuous maps

# Historical note

- The sheaf theory literature before 2015 mostly treats:

  sheaves ON <u>abstract topological spaces</u> OF <u>vectors</u>

Serves pure mathematicians, but no one else, sadly

Algebraic; good for algorithms, but not able to handle noise very well

- The traditional "tool" is *sheaf cohomology*, an algebraic invariant
  - Studies the models in the **absence of data**
  - **Not** noise tolerant
  - Computationally burdensome until very recently (software releases imminent, but still to come!)

# Our discussion today

- The sheaf theory literature before 2015 mostly treats:

  sheaves ON abstract topological spaces OF vectors

Serves pure mathematicians, but no one else, sadly

Algebraic; good for algorithms, but not able to handle noise very well

- Our approach:

  sheaves ON a partial order OF metric spaces

  – Handles both models & data, separately or together
  – **Provably** noise tolerant
  – Computationally more efficient (with caveats)

# Learning objectives for today

Topology provides a handhold for diagrammatic/category-theory modeling:

- Learn to encode various problems as sheaves
  - Some will have "standard" solutions; some won't
- Derive practical solutions from these sheaves
  - Use PySheaf to get numerical estimates!
- Measure, localize, and interpret the extent of consistency within a model with respect to observations

# Starting point...

- A *sheaf* is a data structure that:
  - Pairs a domain of validity with a corresponding model, and

    Graph node: *cell* — Distance metric: *stalk*
  - Explains how the model changes with the domain

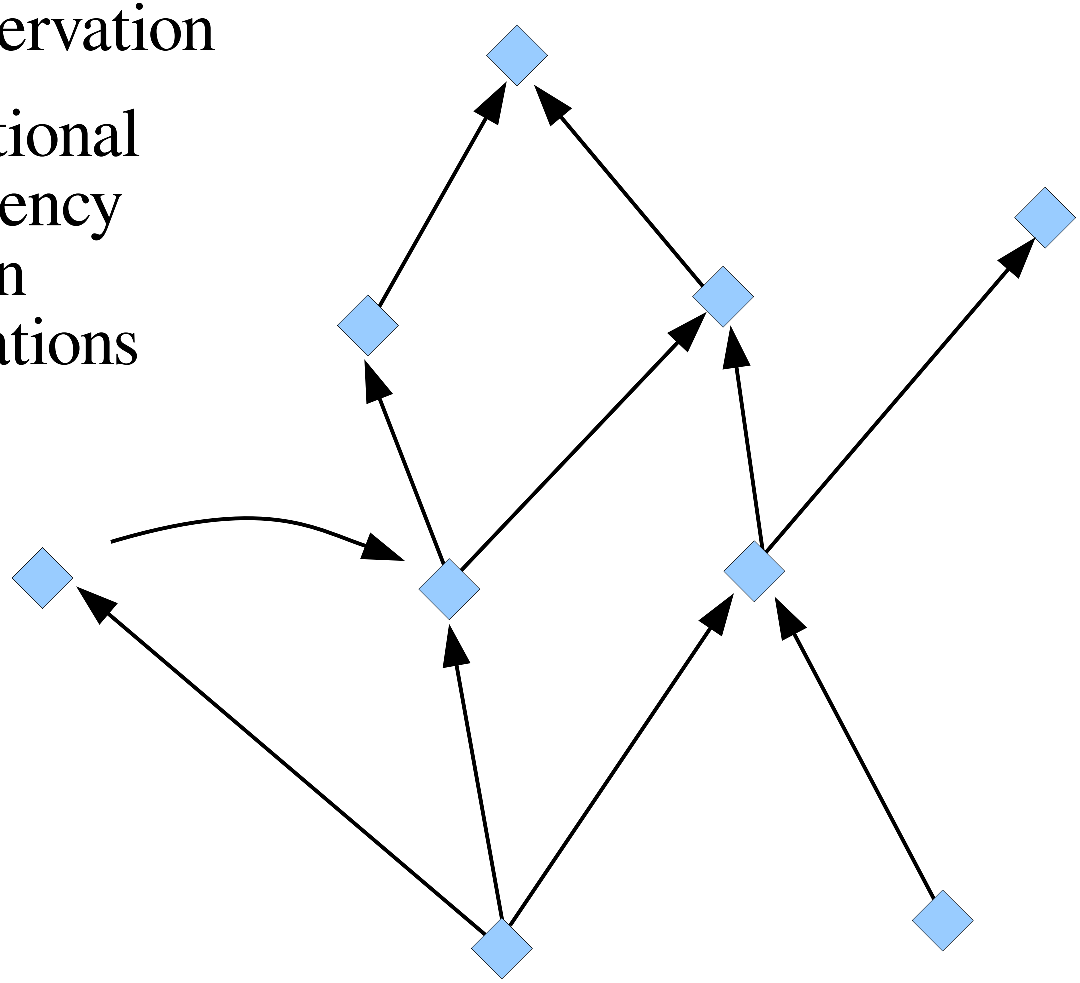
    Function on each graph edge: *restriction*
- An *assignment* is some data "within" the sheaf
- *Consistency radius* measures data-model fit

Let's be a little more precise about what these mean

# Partial order of data sets

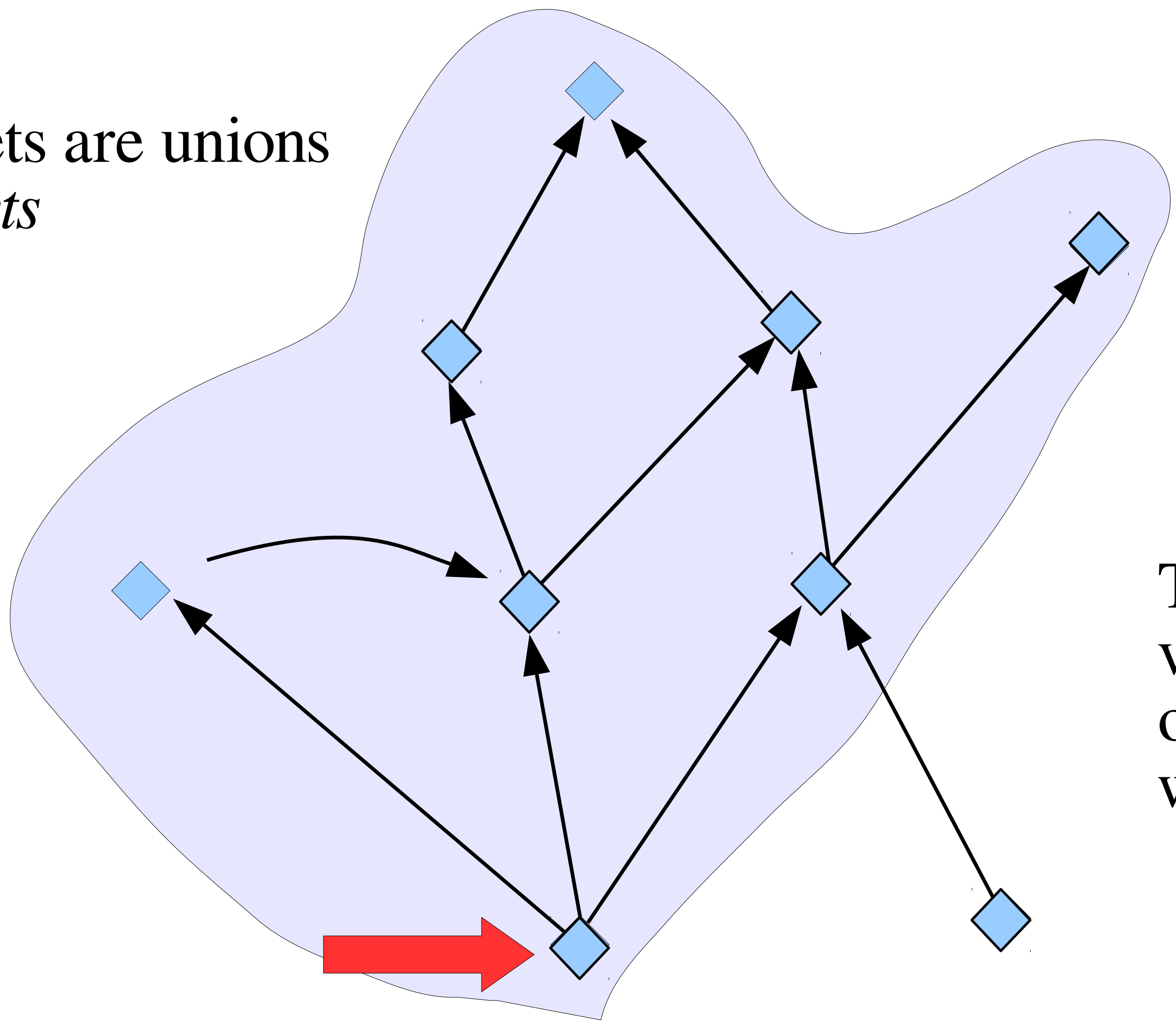
◆ An observation

→ A functional dependency between observations



# Topologizing a partial order

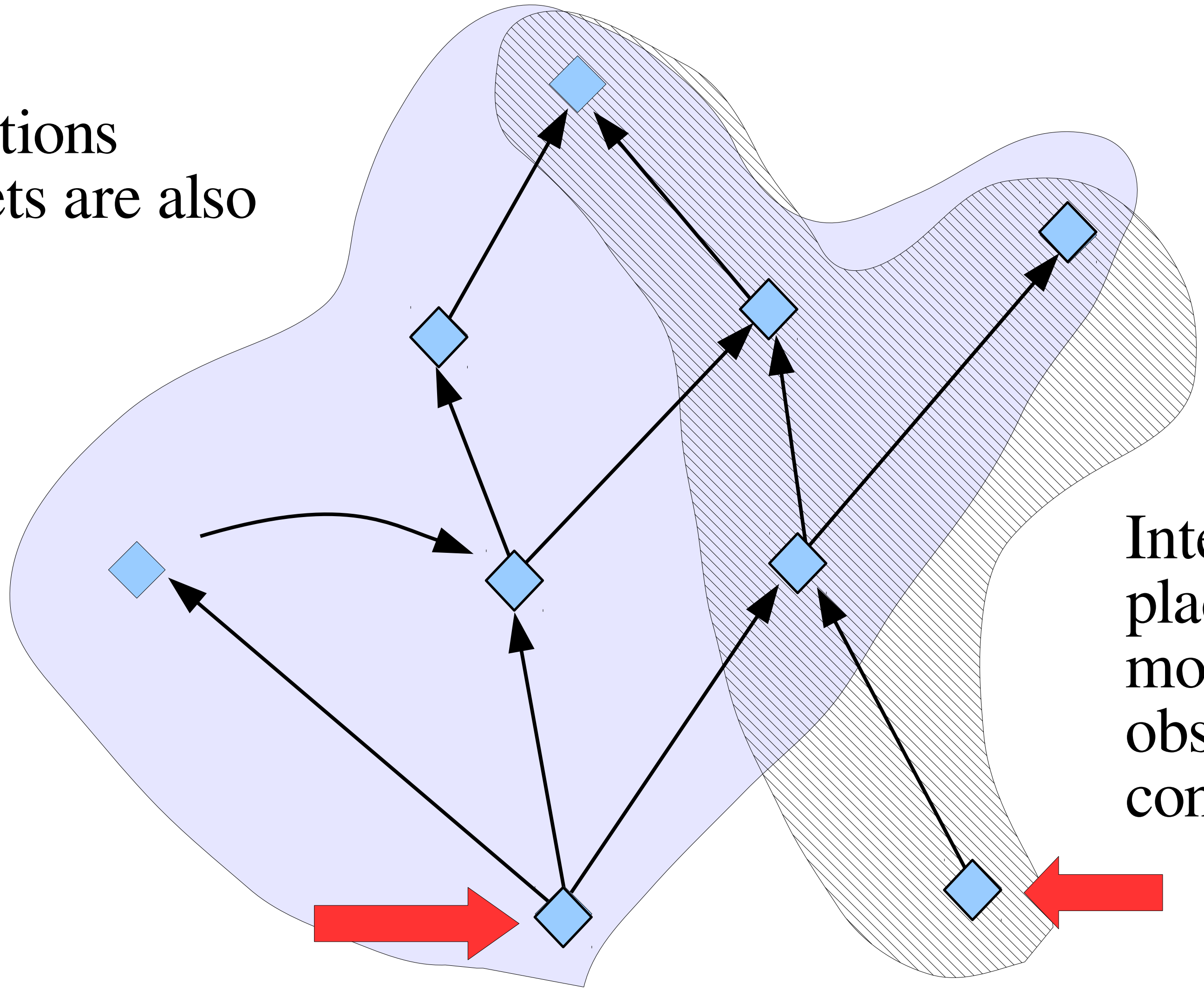
Open sets are unions of *up-sets*



The domain of validity for the observation marked with the arrow

# Topologizing a partial order

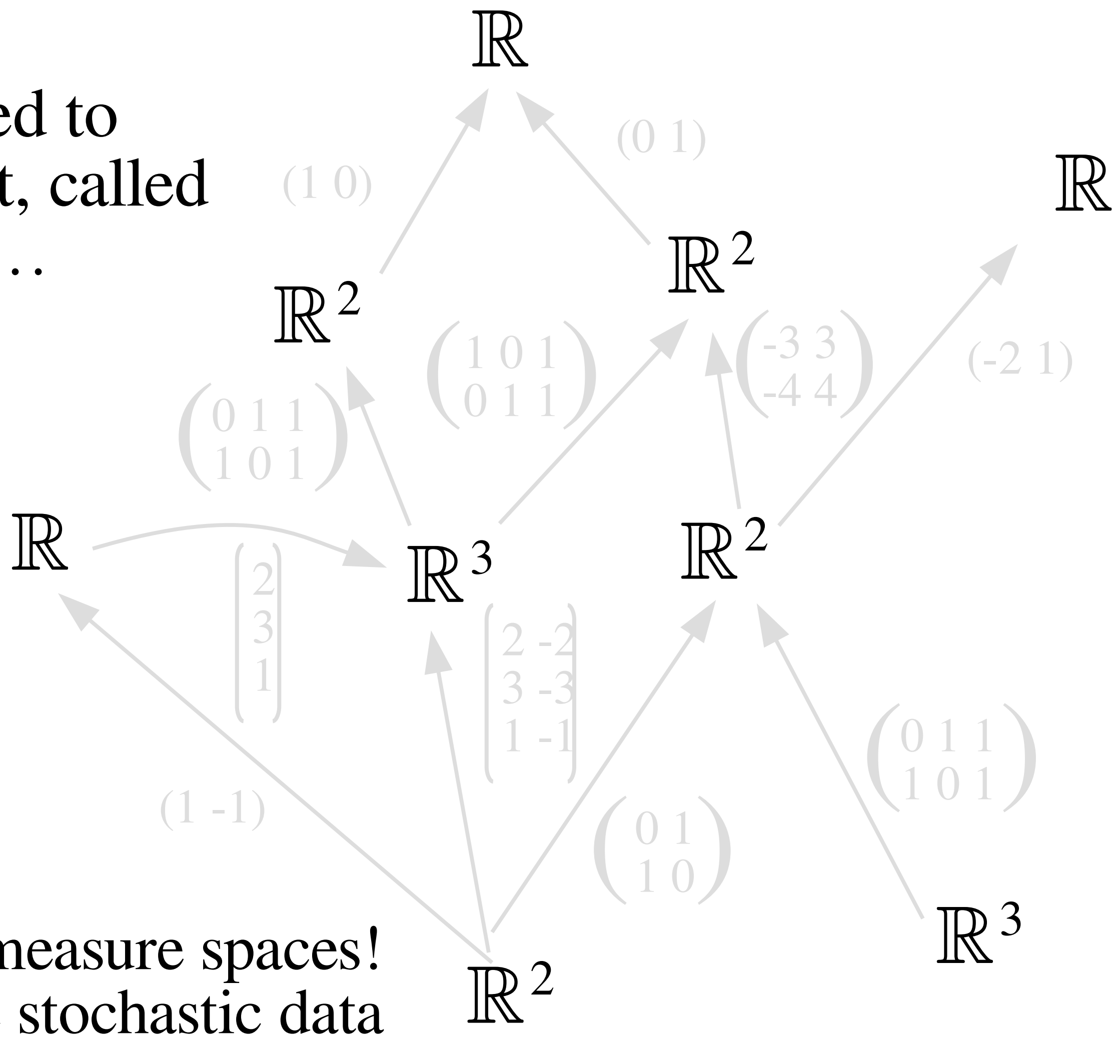
Intersections of up-sets are also up-sets



Intersections are places where two models and their observations might conflict

# A *sheaf* on a poset is...

A set assigned to each element, called a *stalk*, and …

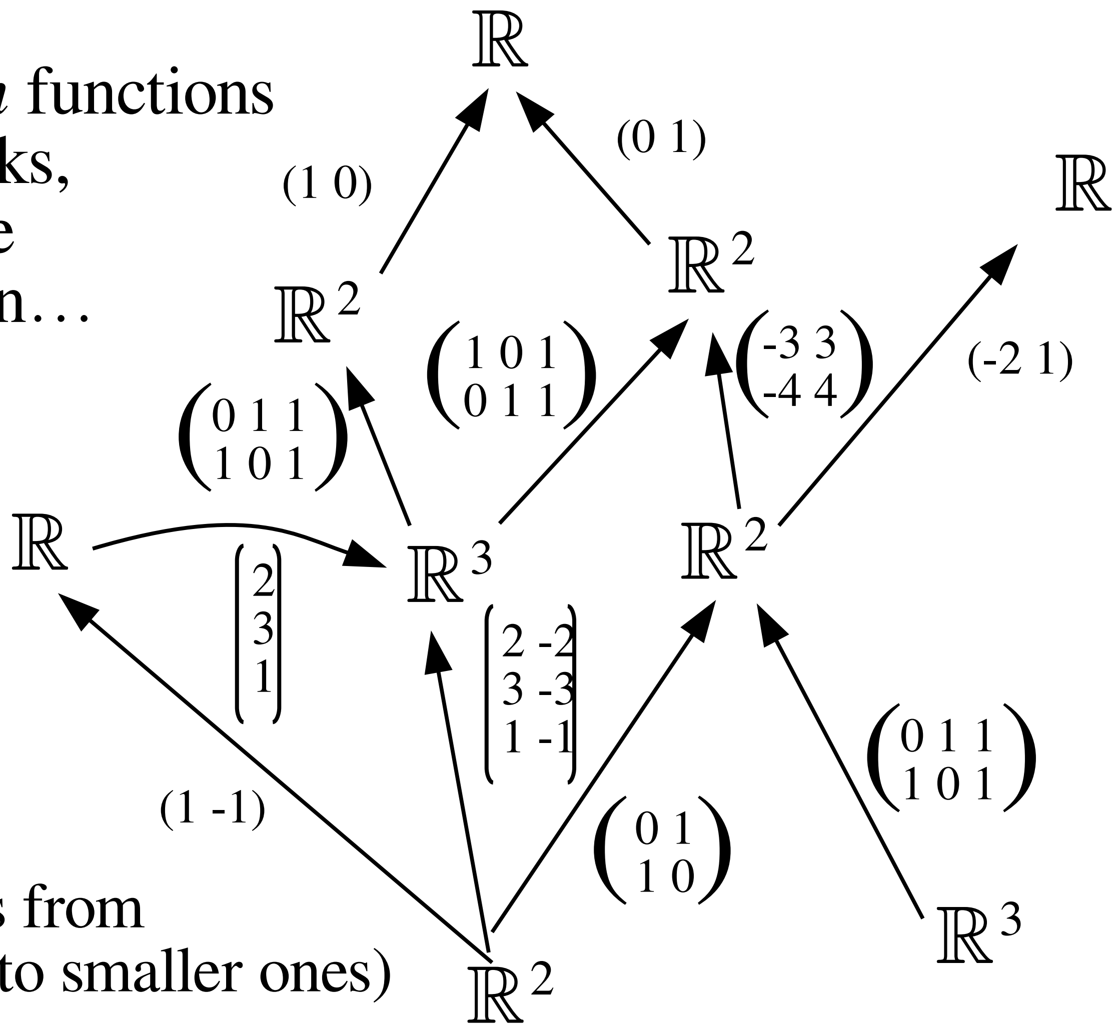



Stalks can be measure spaces!
We can handle stochastic data

This is a *sheaf* **of** vector spaces **on** a partial order

# A *sheaf* on a poset is...

… *restriction* functions between stalks, following the order relation…

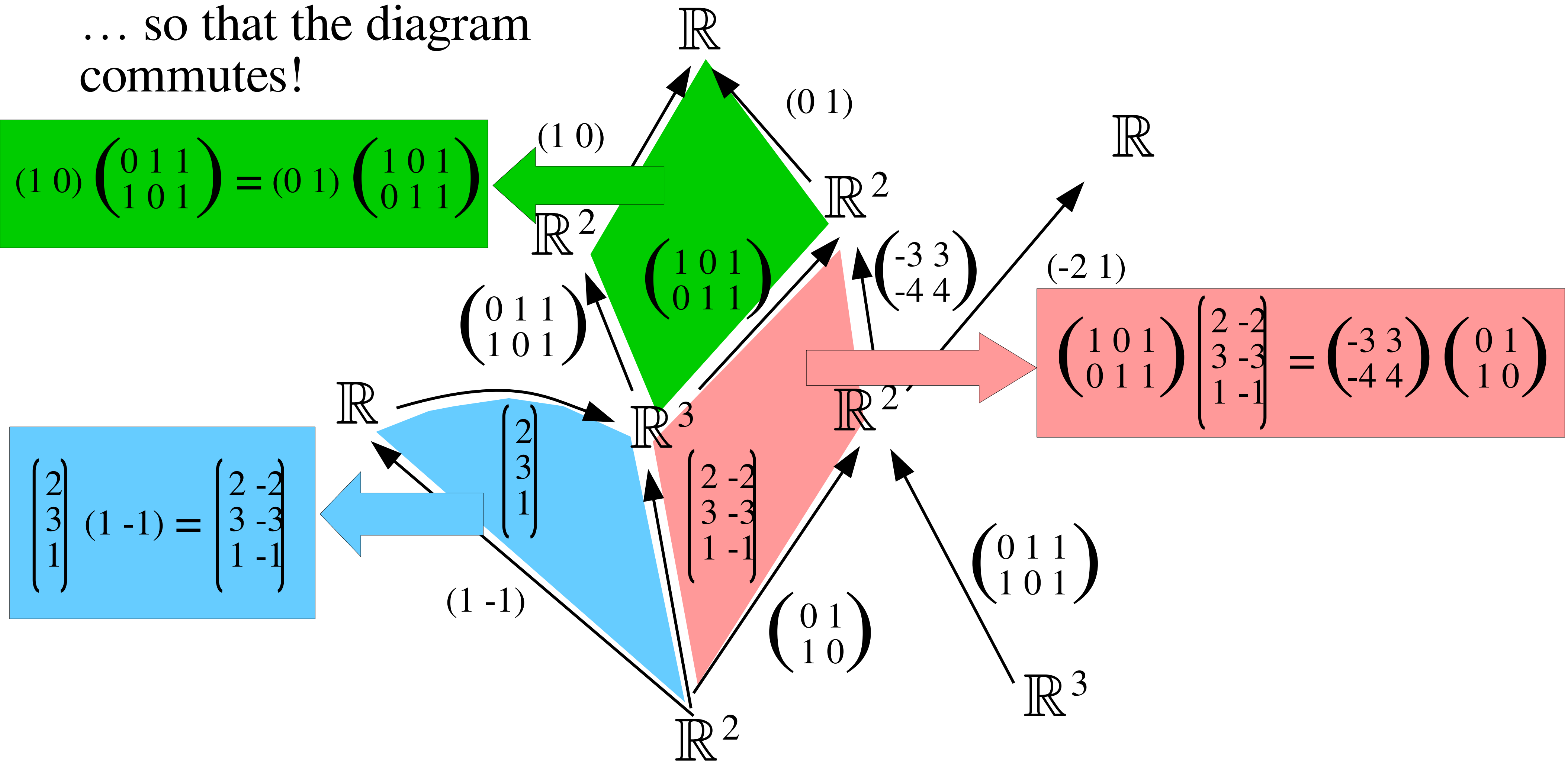



(“Restriction” because it goes from bigger up-sets to smaller ones)

This is a *sheaf* **of** vector spaces **on** a partial order

# A *sheaf* on a poset is...

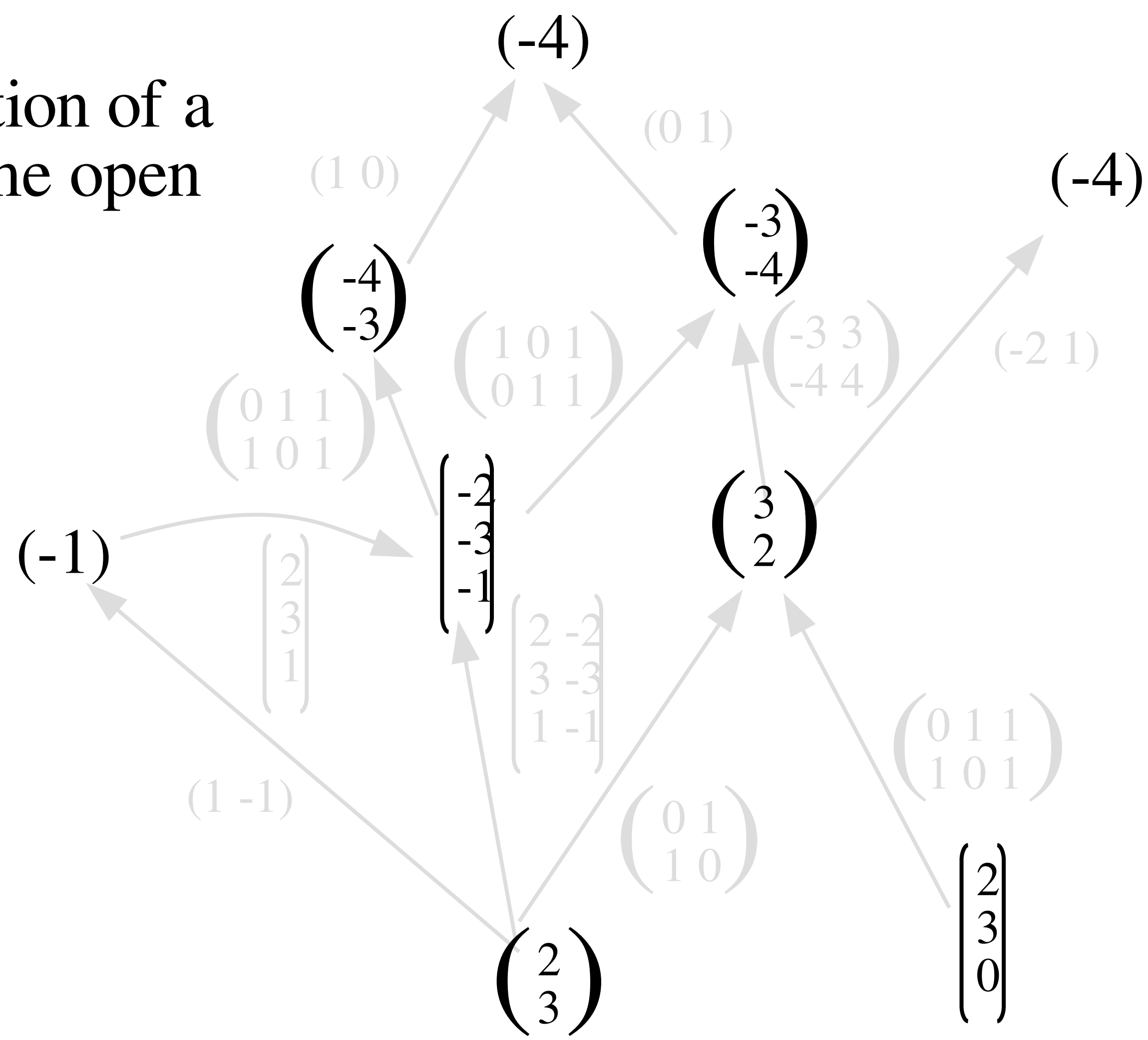
… so that the diagram commutes!




This is a *sheaf* **of** vector spaces **on** a partial order

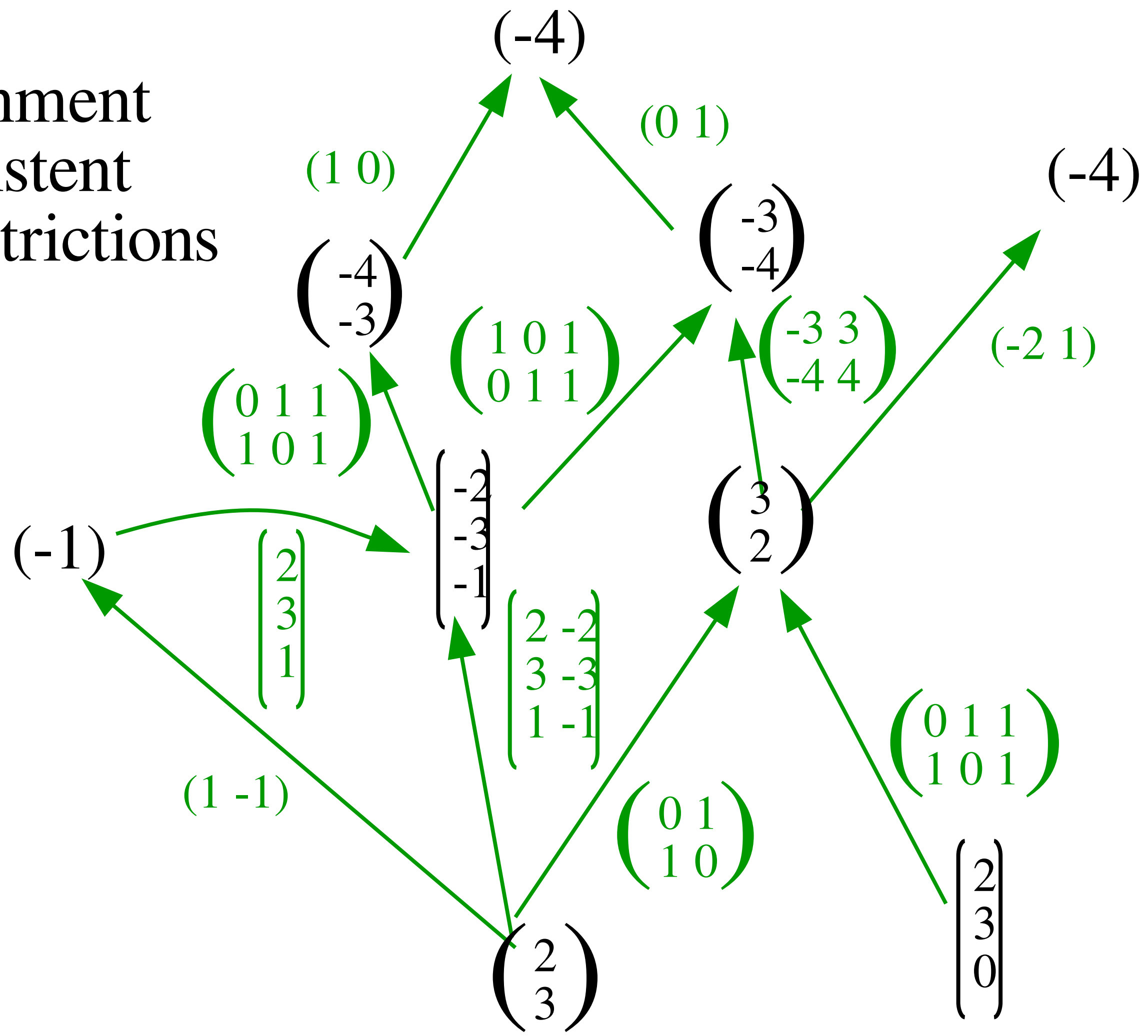
# An *assignment* is...

… the selection of a value on some open sets



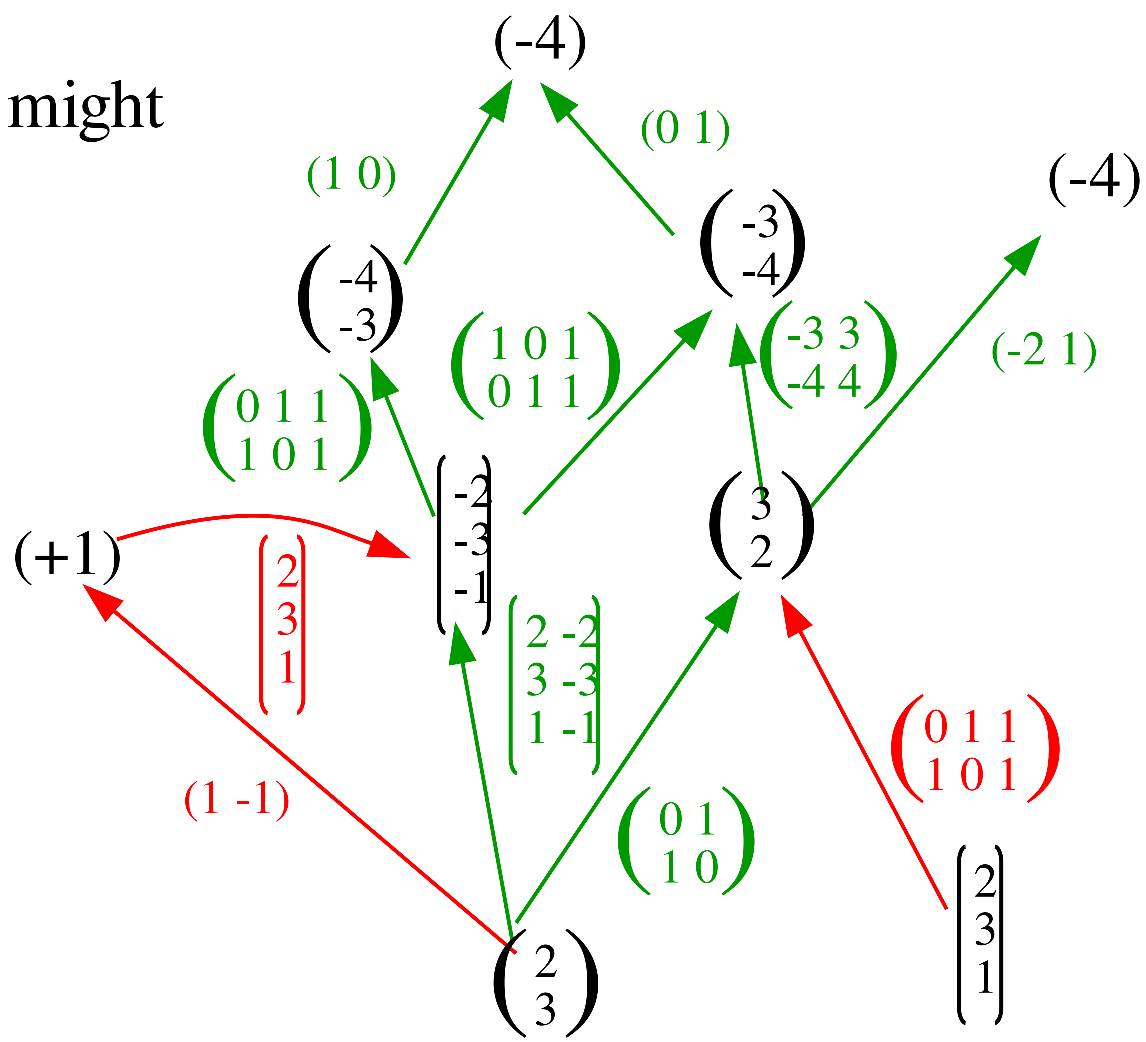
# A *global section* is...

… an assignment that is consistent with the restrictions



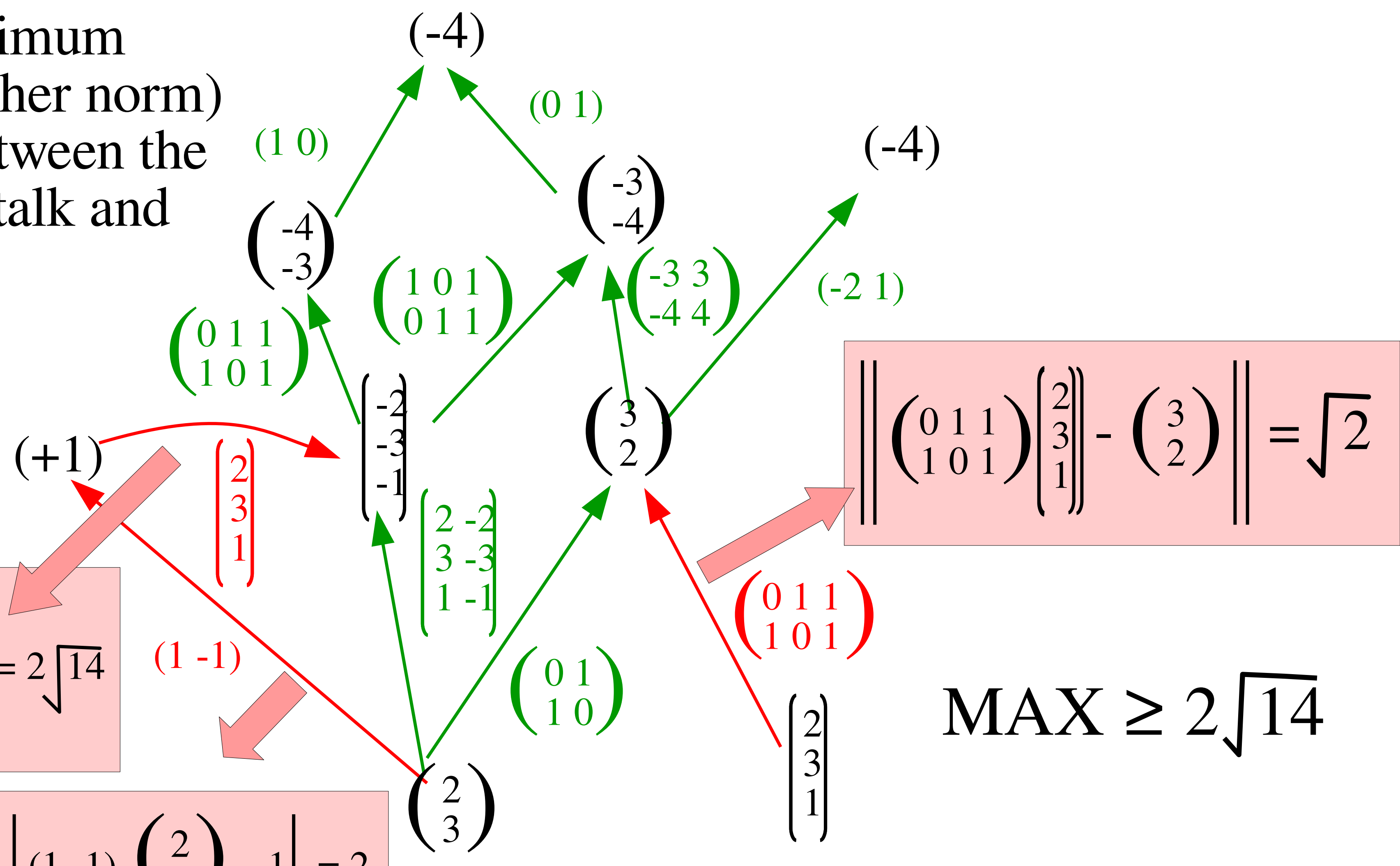
# Some assignments aren't consistent

… but they might be partially consistent



# *Consistency radius* is...

… the maximum (or some other norm) distance between the value in a stalk and the values propagated along the restrictions




$$\left\| \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} - \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right\| = \sqrt{2}$$

$$\left\| \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} (+1) - \begin{bmatrix} -2 \\ -3 \\ -1 \end{bmatrix} \right\| = 2\sqrt{14}$$

$$\left| (1\ {-1}) \begin{pmatrix} 2 \\ 3 \end{pmatrix} - 1 \right| = 2$$

$$\text{MAX} \geq 2\sqrt{14}$$

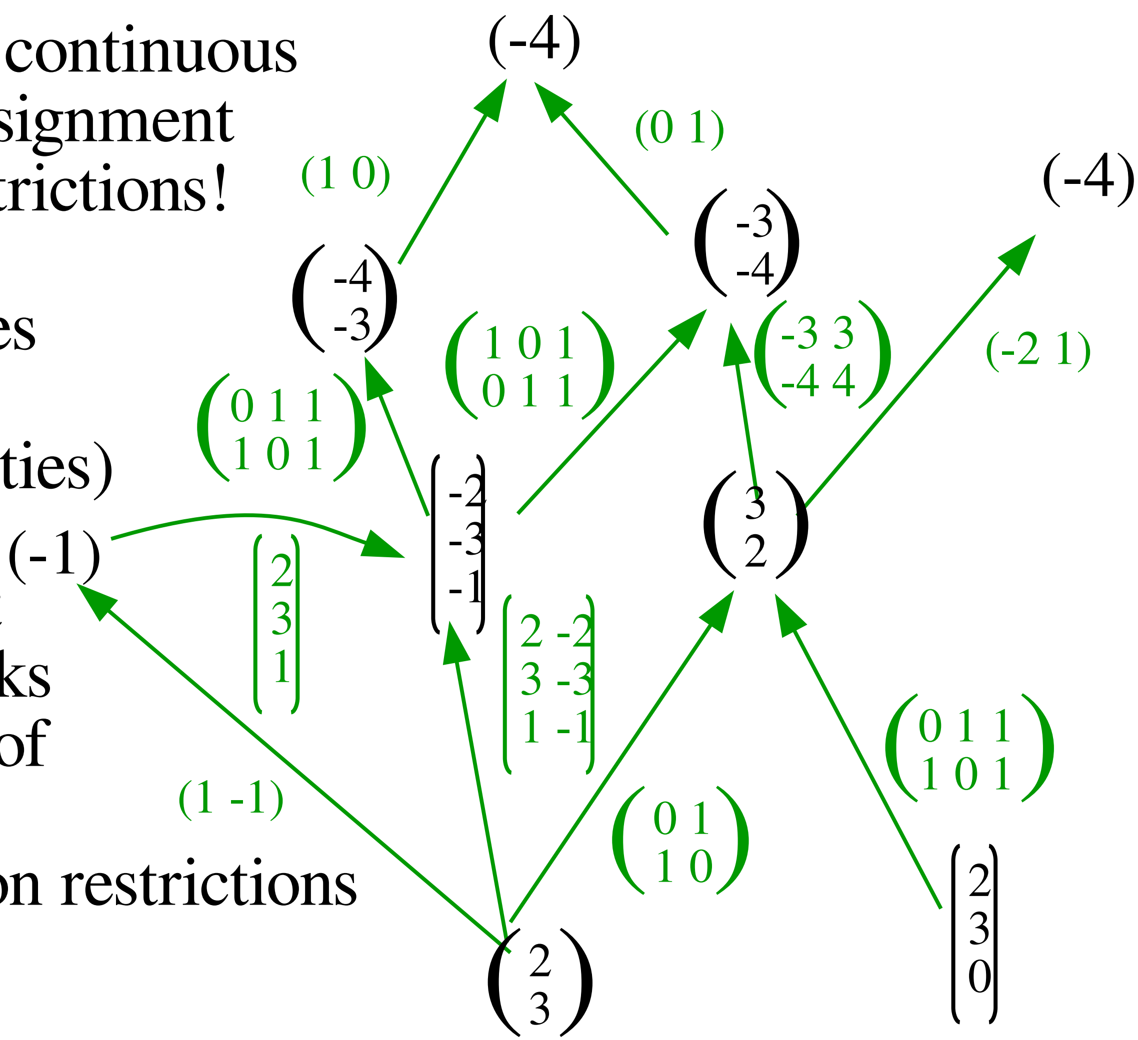
Note: lots more restrictions to check!

# Consistency radius is continuous

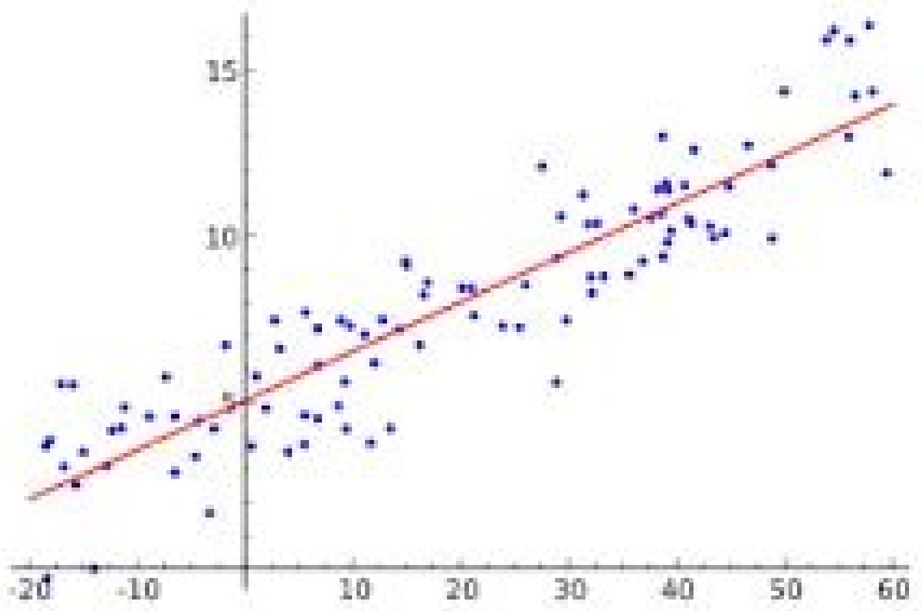
Theorem: It's continuous both in the assignment and in the restrictions!

(Proof involves establishing some inequalities)

Use a product metric on stalks and topology of uniform convergence on restrictions

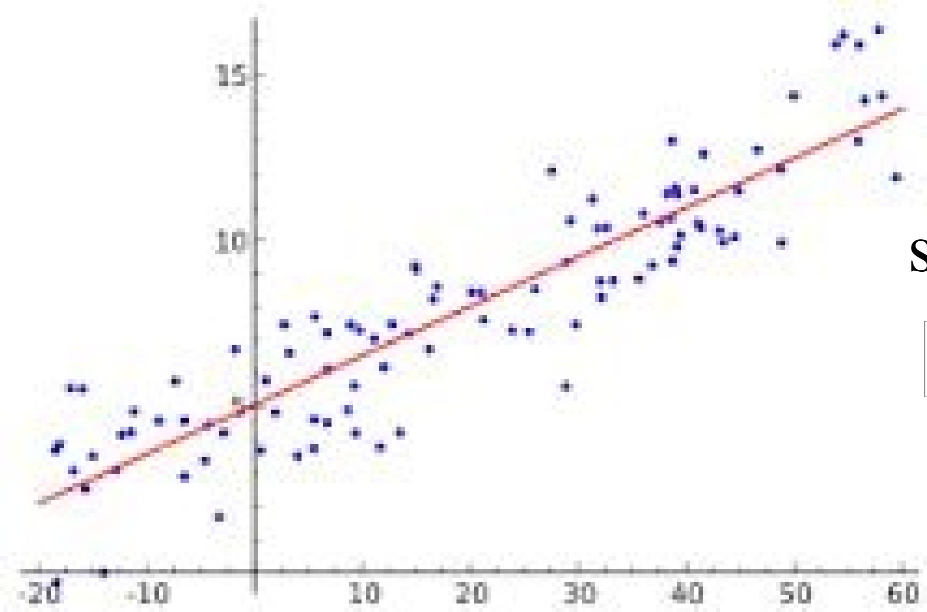


# Consistency radius = aggregated residuals

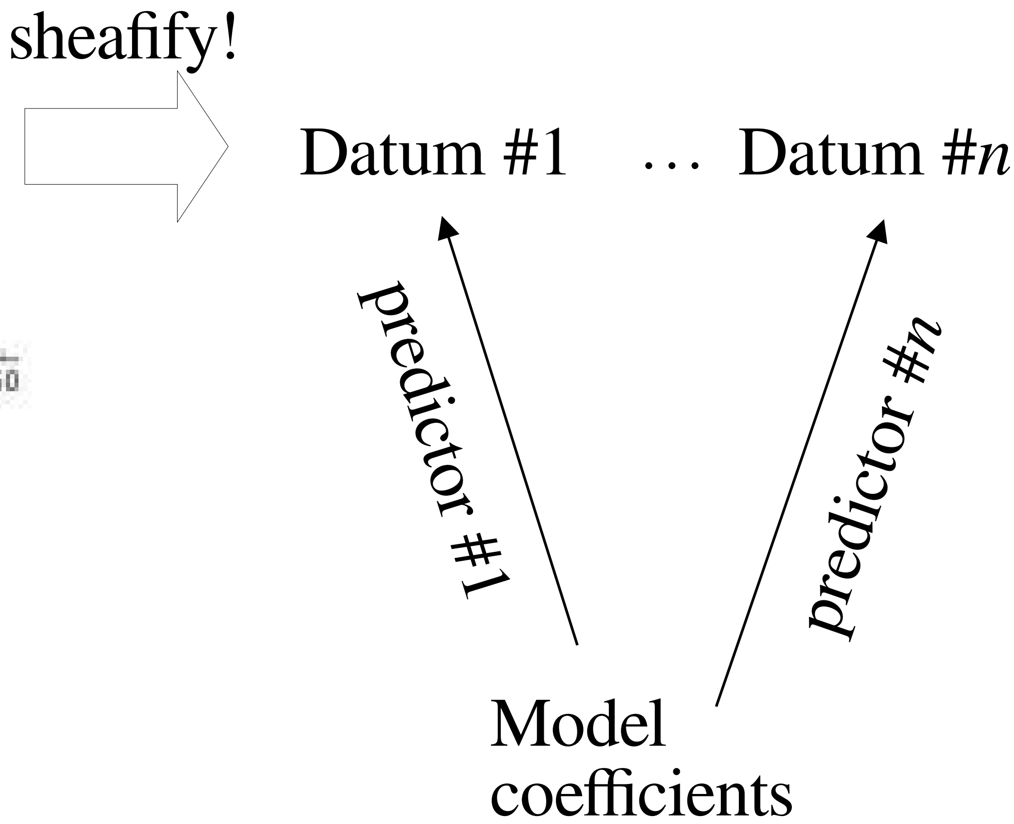



… yes this thing!

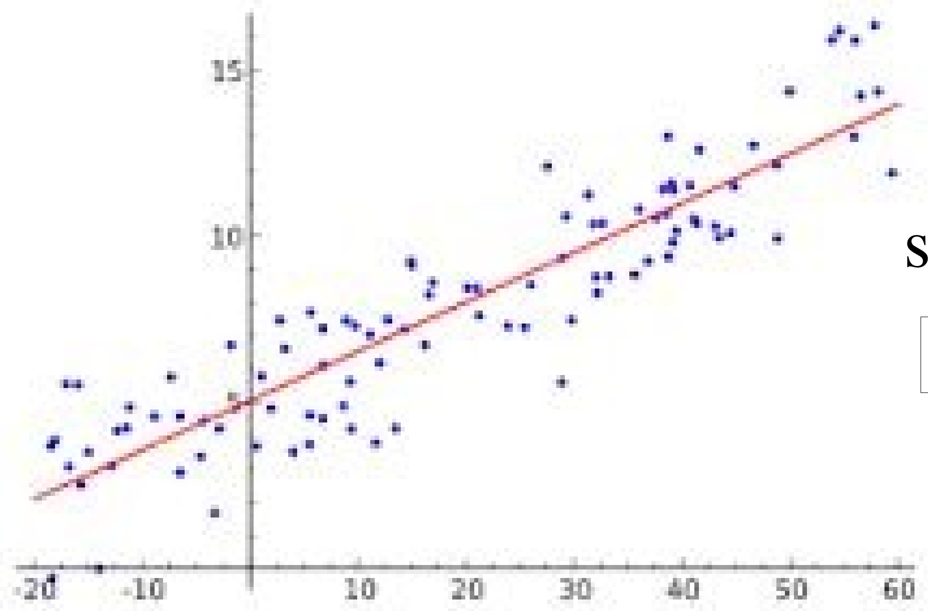
# Linear regression…




… yes this thing!

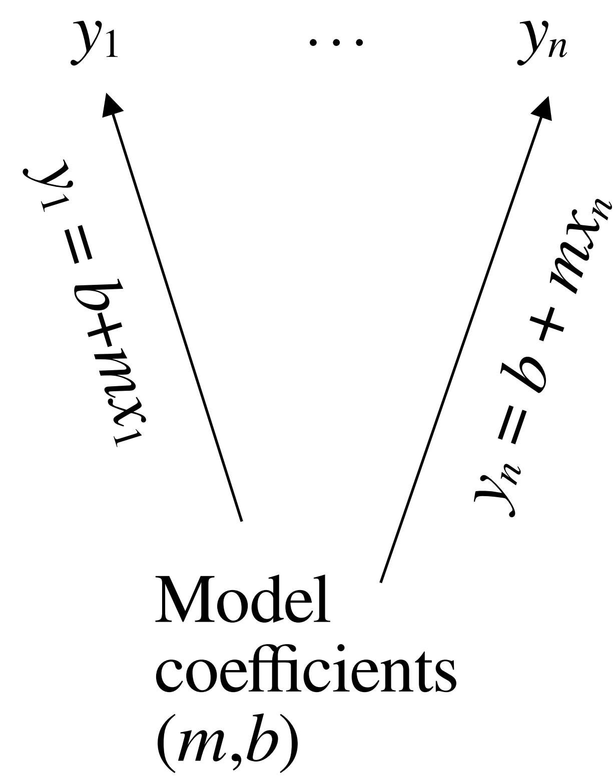


# Linear regression…

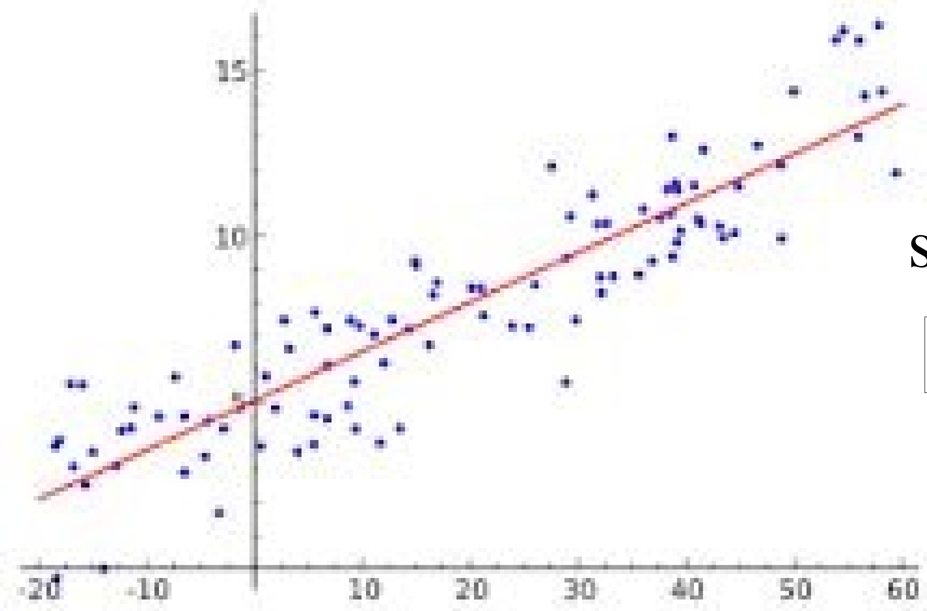



… yes this thing!

sheafify!



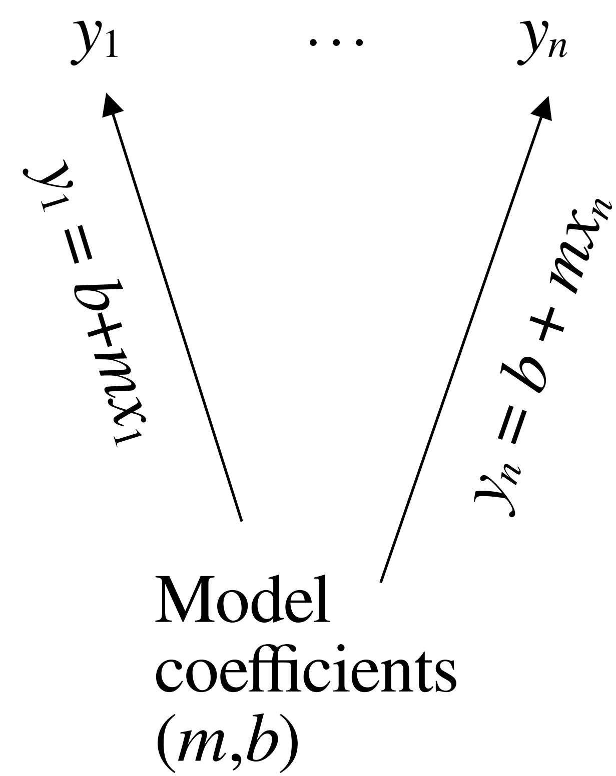
# Linear regression…




… yes this thing!

sheafify!

Tip: Use spans!



# Software!

- Regression as sheaf

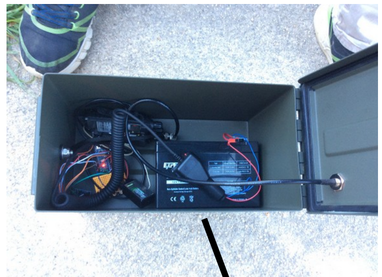
https://colab.research.google.com/drive/1o7N_yQy4QdcUBq48pYzUbUauFVfZVDPp

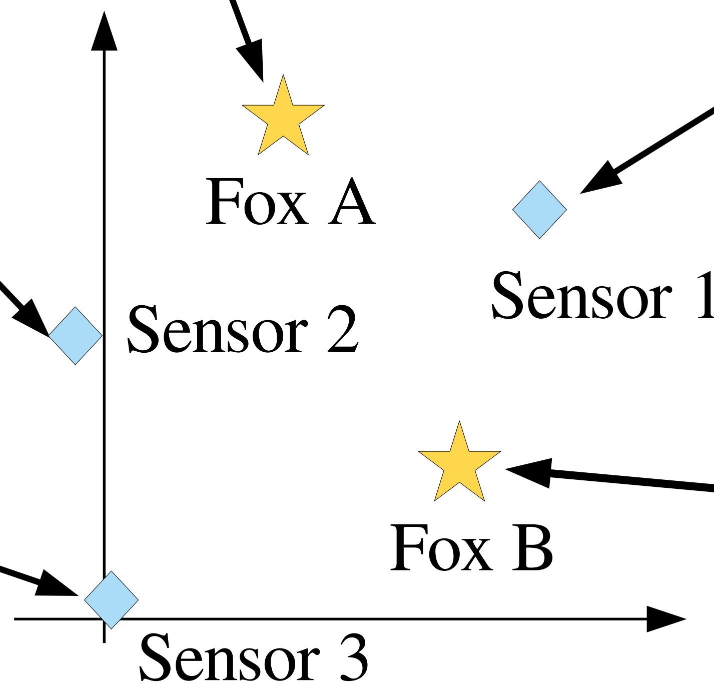
- Radio foxhunting

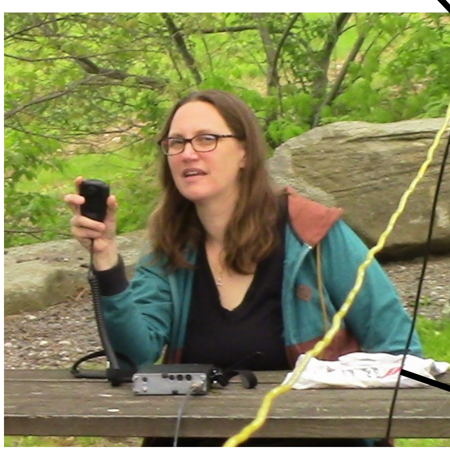
https://colab.research.google.com/drive/16DA4ZEJpgij1paD8eAS8S6-m5pDavntr

# Amateur radio foxhunting

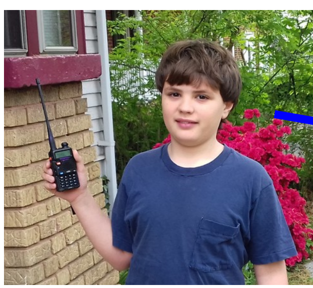
Typical sensors:
- Bearing to Fox
- Fox signal strength
- GPS location

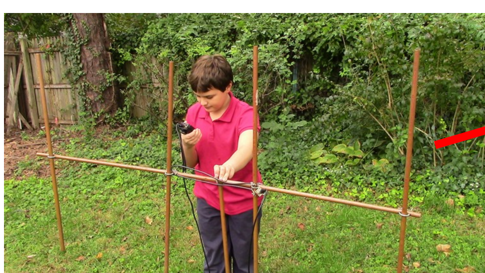


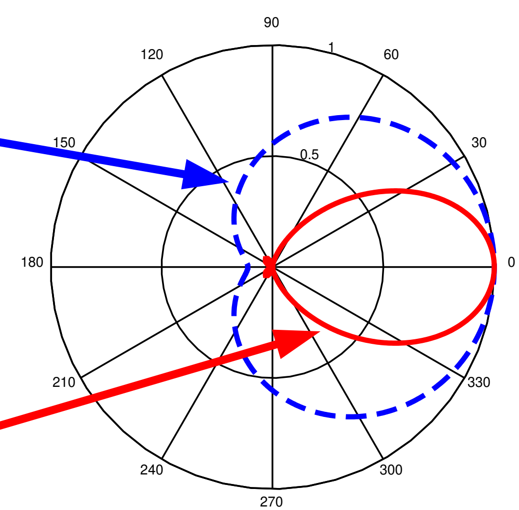


# Bearing sensors

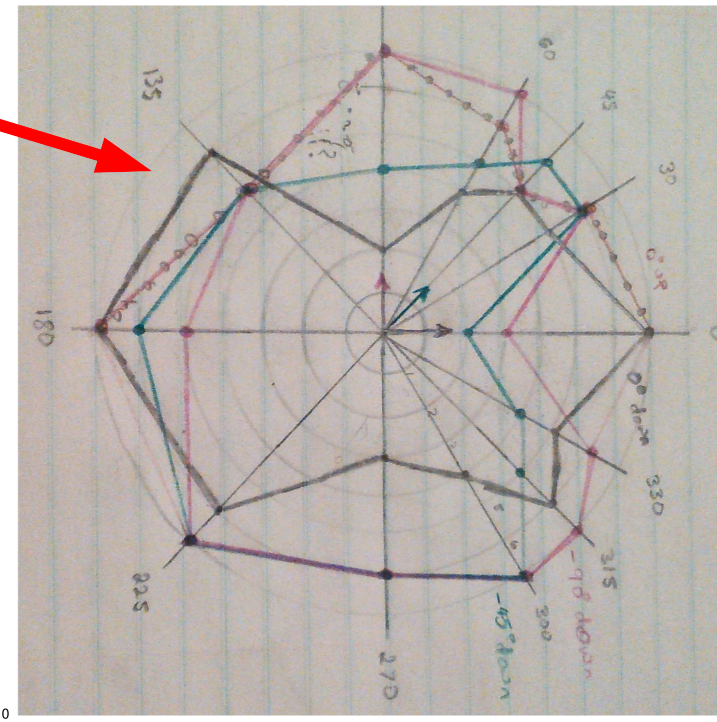


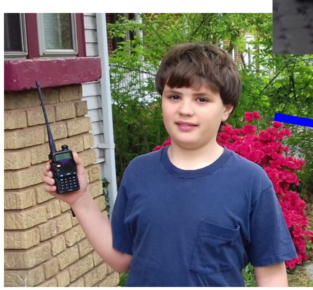


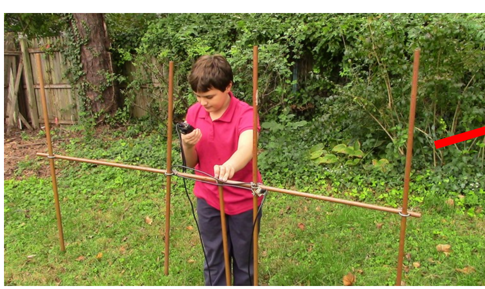


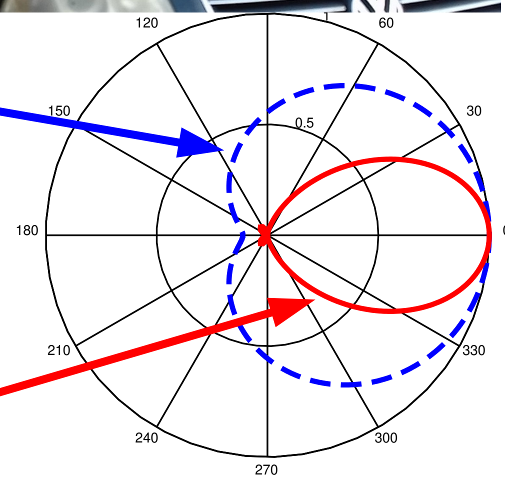

Antenna pattern

# Bearing sensors… reality…









Antenna pattern

# Bearing sheaf

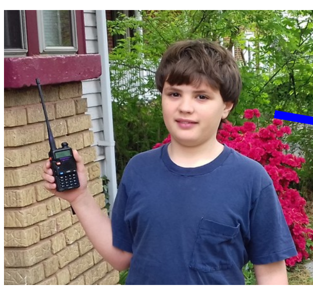
**Fox position** **Sensor position, Bearing**

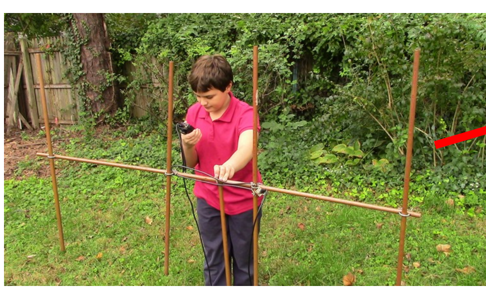
$$\mathbb{R}^2 \xleftarrow{\text{pr}_1} \mathbb{R}^2 \times \mathbb{R}^2 \xrightarrow{(\text{pr}_2,\, M_{bearing})} \mathbb{R}^2 \times S^1$$

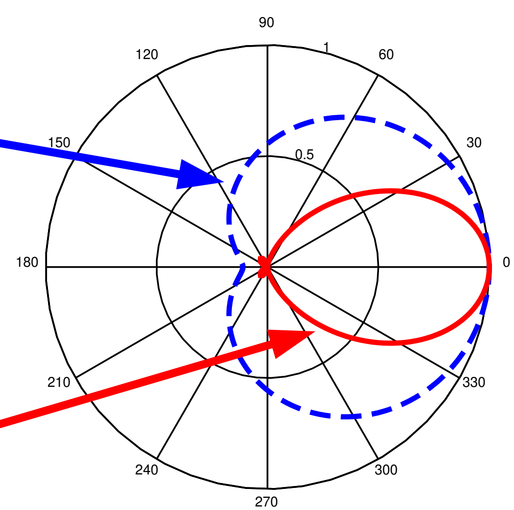
True observations

Virtual observations

Fox position, Sensor position








Antenna pattern

# Bearing sheaf (two sensors)

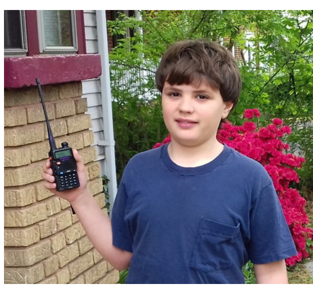
**Sensor 1 position, Bearing** **Fox position** **Sensor 2 position, Bearing**

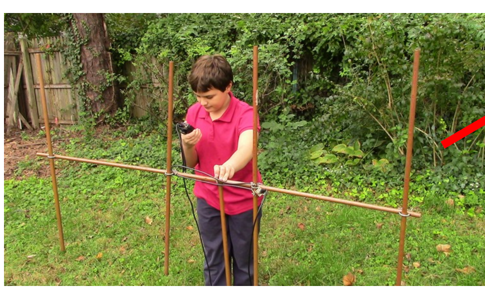
$\mathbb{R}^2 \times S^1$ $\quad$ $\mathbb{R}^2$ $\quad$ $\mathbb{R}^2 \times S^1$

$(\mathrm{pr}_2, M_{bearing})$ $\quad$ $\mathrm{pr}_1$ $\quad$ $\mathrm{pr}_1$ $\quad$ $(\mathrm{pr}_2, M_{bearing})$

$\mathbb{R}^2 \times \mathbb{R}^2$ $\quad$ $\mathbb{R}^2 \times \mathbb{R}^2$

Fox position, Sensor 1 position $\quad$ Fox position, Sensor 2 position

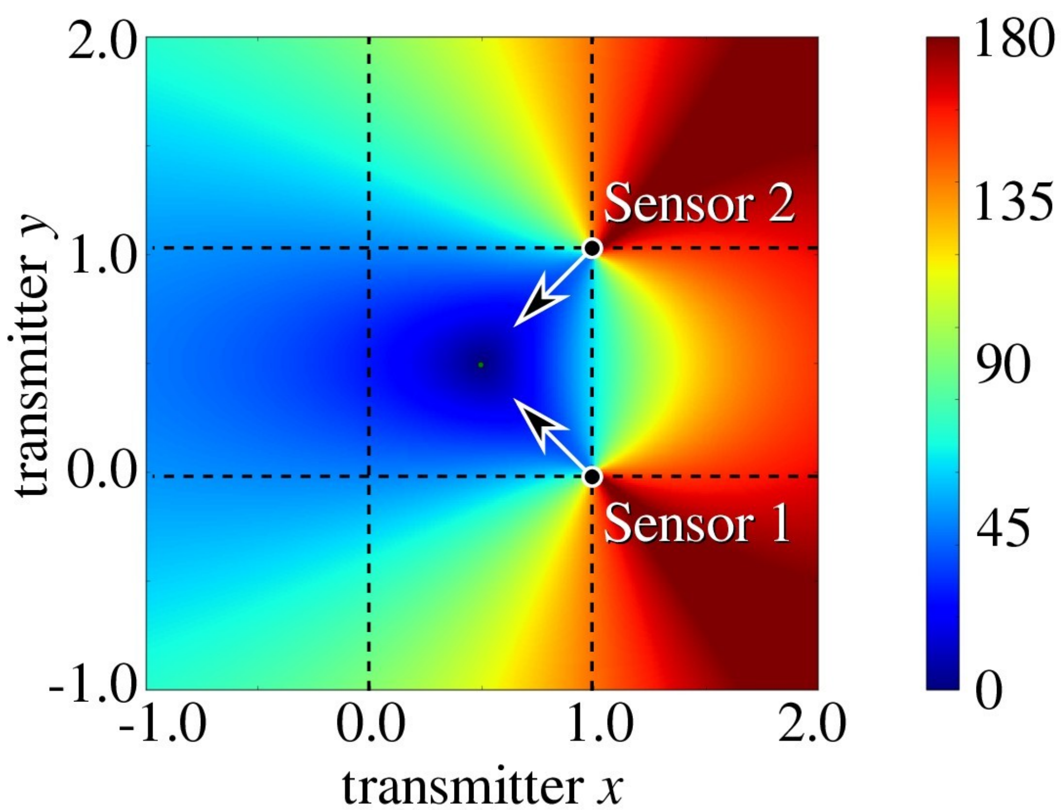


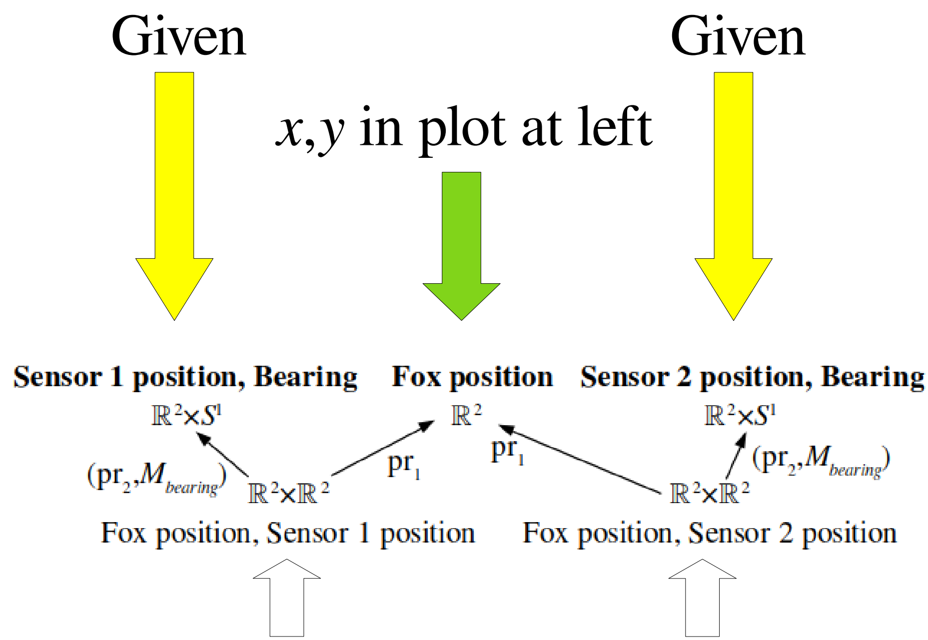


Global sections of this sheaf correspond to two bearings whose sight lines intersect at the fox transmitter

# Consistency of proposed fox locations

Consistency radius minimization …



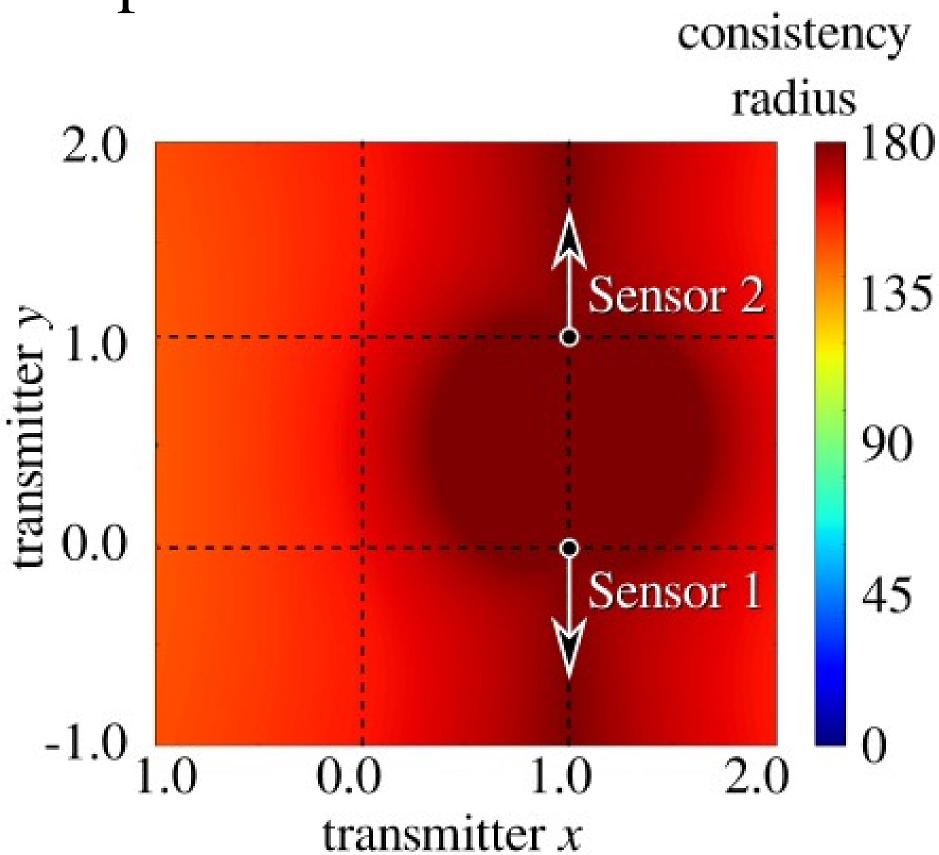




… converges to a likely fox location

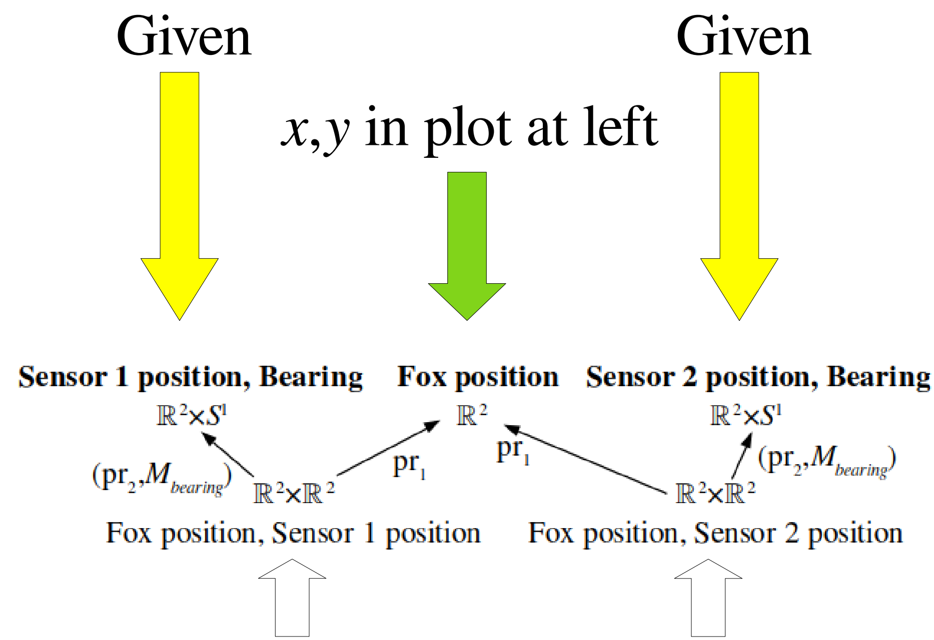
Inferred by minimum consistency radius (closed form solution in this case)

# Consistency of proposed fox locations

An impossible situation...

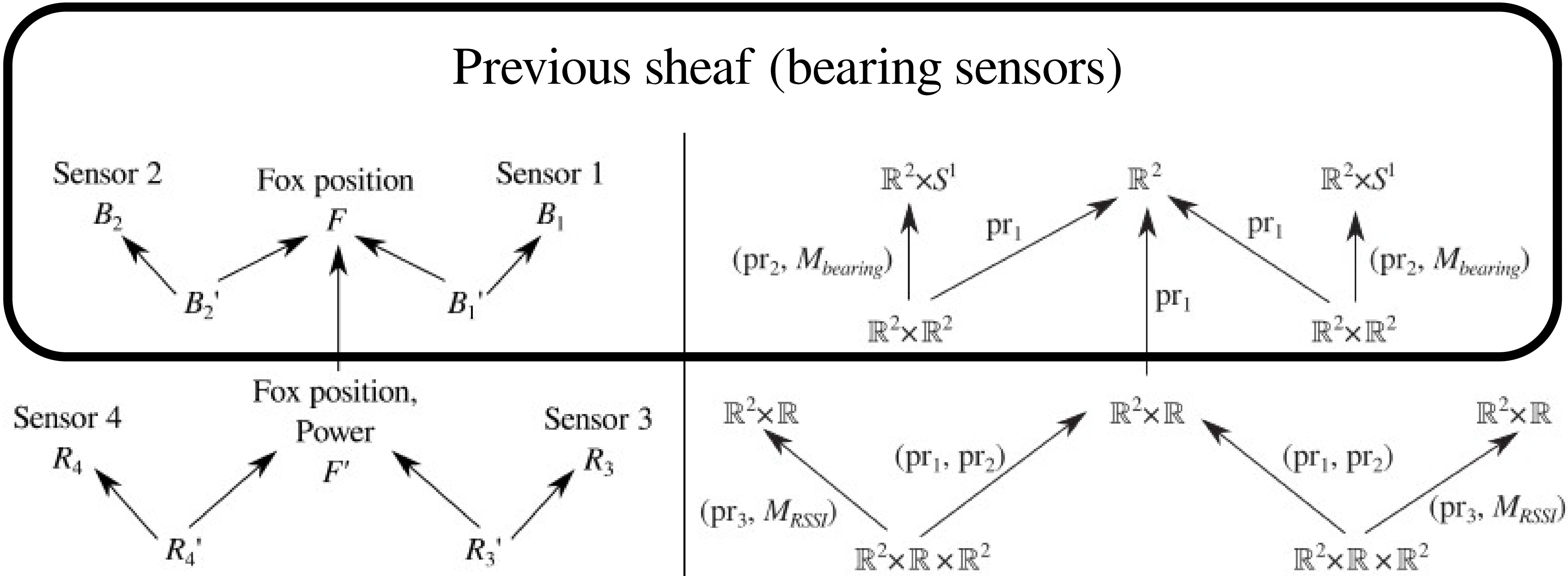



… does not converge!




Inferred by minimum consistency radius (closed form solution in this case)

# A larger sheaf from more sensors

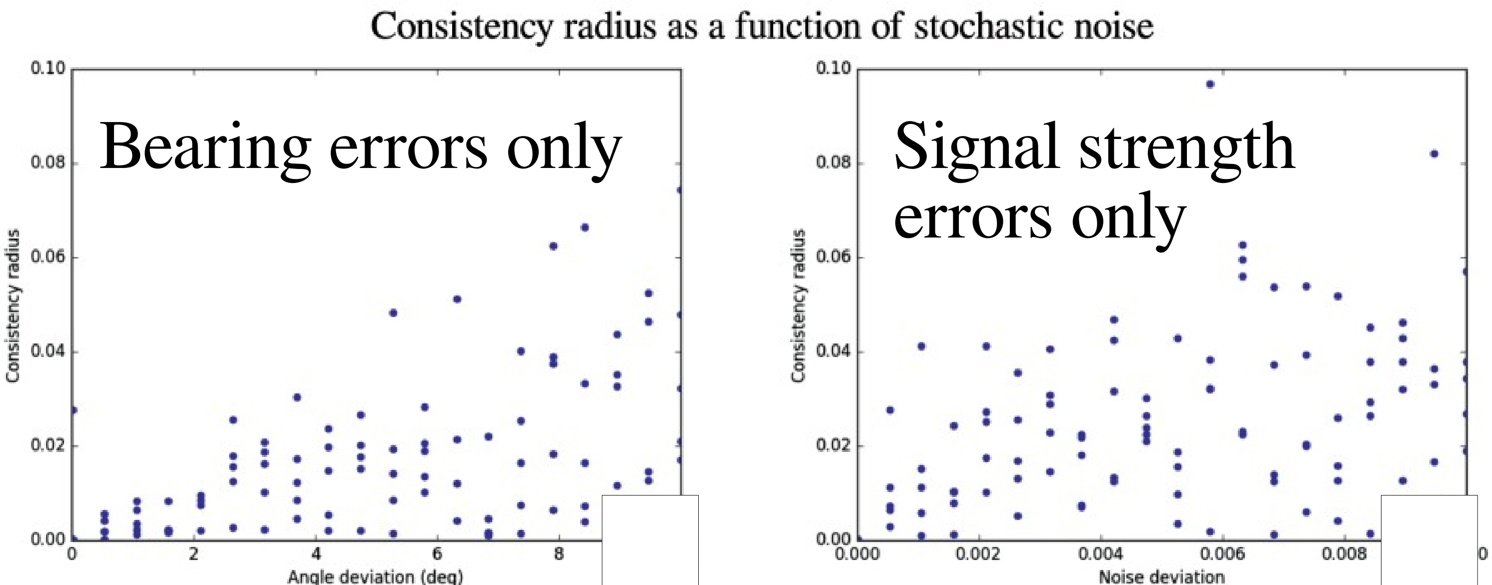



Sensors & observations

Stalks & restrictions

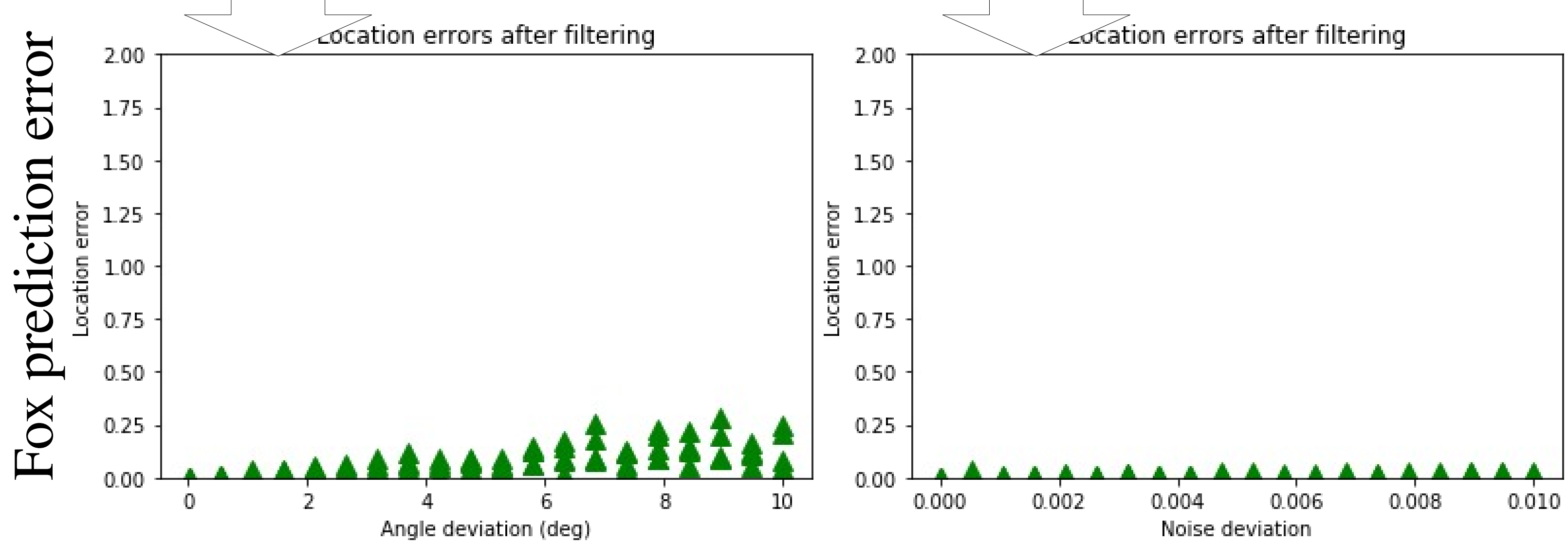
This larger sheaf contains bearing and signal strength sensors

# Consistency radius tracks noise level




all cells with unknown values were randomized

Optimization finds the fox!



# Interpretation

- Sheaf: a **data structure** for modeling consistency
- Assignment: an **instance** of the data housed in a sheaf
- Consistency radius: **how well** do data and model agree?
- Consistency radius optimization: **predict** some missing or less-noisy data

# We've now seen...

- A few examples of sheaf models for a few problems
- How do we build sheaf models in general?

# A differential equation example

- Consider $u' = f(u)$ on the real line
- $C^k(\mathbb{R},\mathbb{R}^d)$ is the space of $k$-times continuously differentiable functions
- The equation might be expressed diagrammatically:

$$\begin{array}{c} u' : C^0(\mathbb{R},\mathbb{R}^d) \\ f \uparrow \qquad \uparrow d/dt \\ u : C^1(\mathbb{R},\mathbb{R}^d) \end{array}$$

# A differential equation example

- Consider $u' = f(u)$ on the real line
- $C^k(\mathbb{R},\mathbb{R}^d)$ is the space of $k$-times continuously differentiable functions
- The equation might be expressed diagrammatically:

$$\begin{array}{c} u' : C^0(\mathbb{R},\mathbb{R}^d) \\ f \uparrow \quad \uparrow d/dt \\ u : C^1(\mathbb{R},\mathbb{R}^d) \end{array}$$

But wait, diagram does not commute, so it cannot be a sheaf!

# A differential equation example

- Consider $u' = f(u)$ on the real line
- $C^k(\mathbb{R},\mathbb{R}^d)$ is the space of $k$-times continuously differentiable functions
- The equation might be expressed diagrammatically:

$$
\begin{array}{c}
u' : C^0(\mathbb{R},\mathbb{R}^d) \\
f \uparrow \qquad\quad \uparrow d/dt \\
u : C^1(\mathbb{R},\mathbb{R}^d)
\end{array}
$$

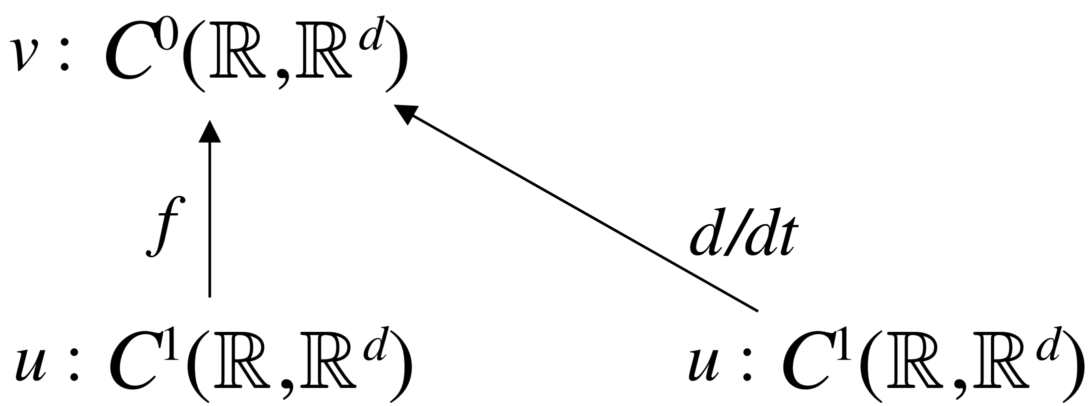
But wait, diagram does not commute, so it cannot be a sheaf!
Well, OK. It **is** a sheaf on the free category gen'd by the graph.
That's awkward*. We are going to stick with posets today.

*I would need to define sieves &c, but won't b/c that is rather subtle

# A differential equation example

A standard trick: replace $u' = f(u)$ with the system:

- $v = f(u)$
- $v = d/dt\,(u)$

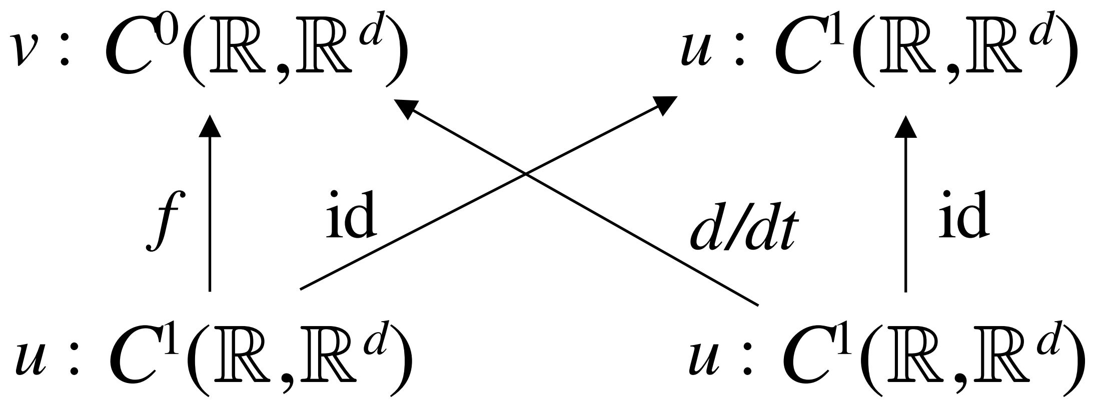



But wait, now the two copies of $u$ don't have to agree...

# A differential equation example

A standard trick: replace $u' = f(u)$ with the system:

- $v = f(u)$
- $v = d/dt\,(u)$




Sections of this sheaf are solutions to the original equation, because this requires all three copies of $u$ to agree

# Multi-equation sheaves

- Theorem: (R.) For every system of equations, there is a sheaf whose global sections are solutions
  - Base poset has two levels: Equations < Variables
  - Stalk over each variable is that variable's set of possible values
  - Stalk over an equation is a subset of the product of the variables involved
  - Restriction maps are projections

Source: M. Robinson, "Sheaf and duality methods for analyzing multi-model systems," arXiv:1604.04647

# Goodwin macroeconomic model

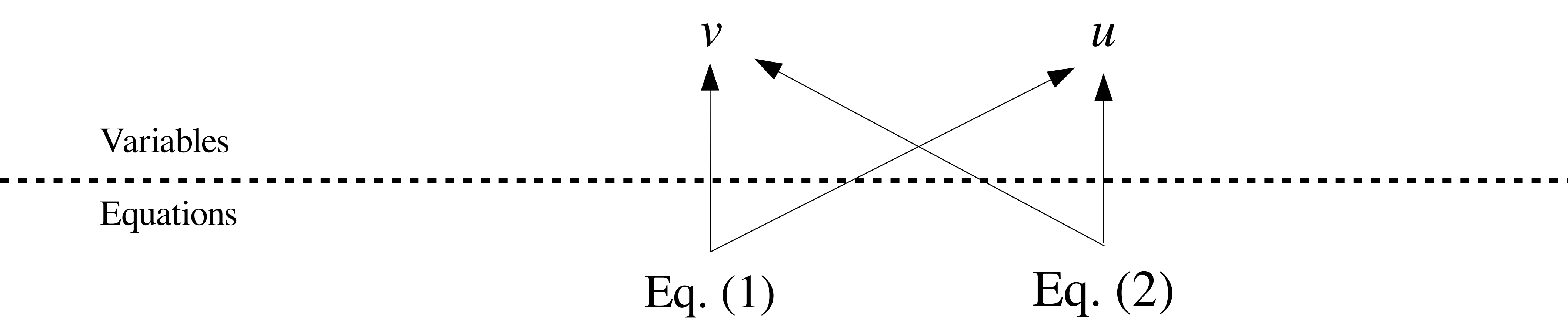
- A simple description of a national economy:

$$\dot{v} = \quad v(t)\left(\frac{1}{\sigma} - (\alpha+\beta) - \frac{u(t)}{\sigma}\right) \qquad (1)$$

$v$ = Employment rate

$$\dot{u} = \quad u(t)\left(-(\alpha+\gamma) + (\rho v(t))\right). \qquad (2)$$

$u$ = Workers' share of income



# Goodwin macroeconomic model

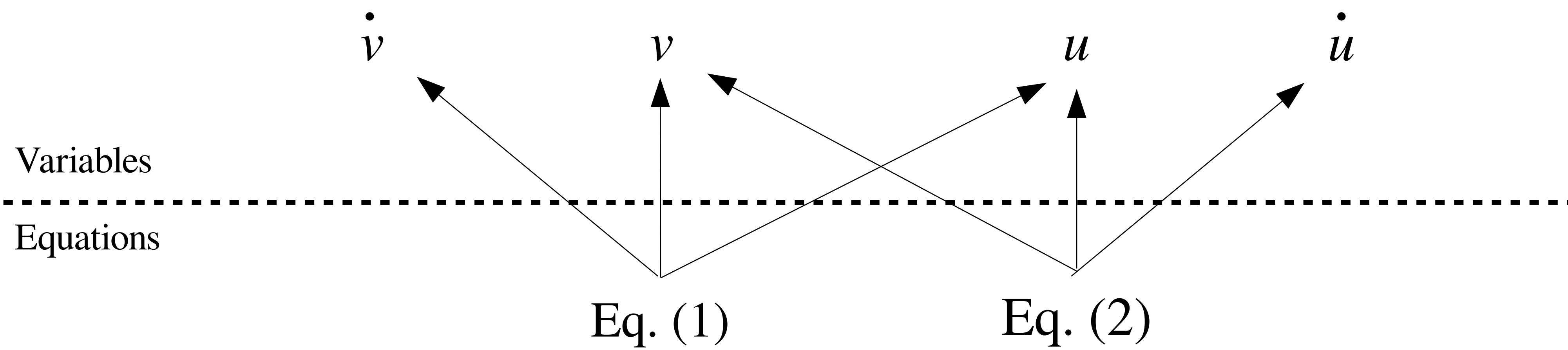
- A simple description of a national economy:

$$\dot{v} = v(t)\left(\frac{1}{\sigma} - (\alpha+\beta) - \frac{u(t)}{\sigma}\right) \quad (1)$$

$v$ = Employment rate

$$\dot{u} = u(t)\left(-(\alpha+\gamma) + (\rho v(t))\right). \quad (2)$$

$u$ = Workers' share of income



# Goodwin macroeconomic model

- A simple description of a national economy:

$$\dot{v} = v(t)\left(\frac{1}{\sigma} - (\alpha+\beta) - \frac{u(t)}{\sigma}\right) \quad (1)$$

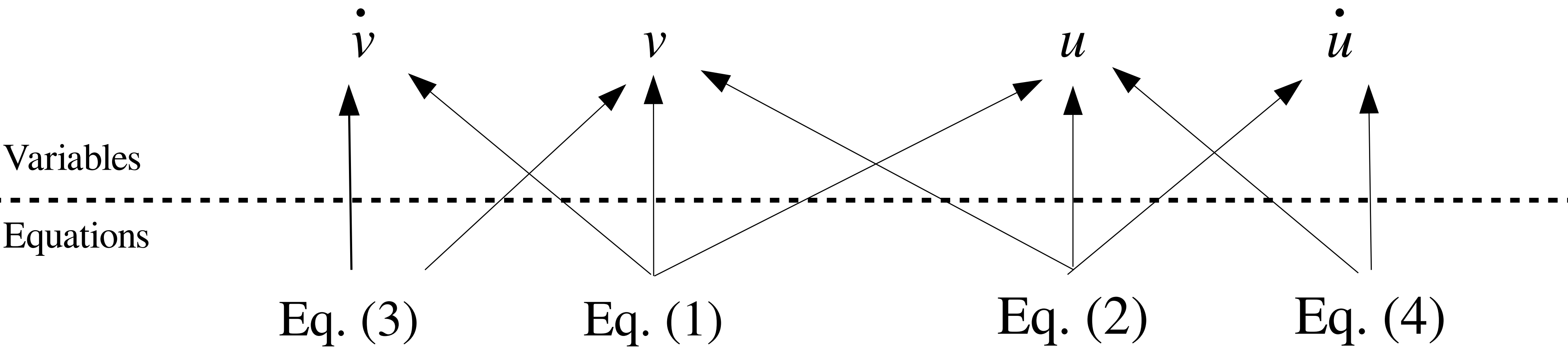
$v$ = Employment rate

$$\dot{u} = u(t)\left(-(\alpha+\gamma) + (\rho v(t))\right). \quad (2)$$

$u$ = Workers' share of income

$$\dot{v} = dv/dt \quad (3)$$

$$\dot{u} = du/dt \quad (4)$$



# Goodwin macroeconomic model

- A simple description of a national economy:

$$\dot{v} = \quad v(t)\left(\frac{1}{\sigma} - (\alpha+\beta) - \frac{u(t)}{\sigma}\right) \qquad (1)$$

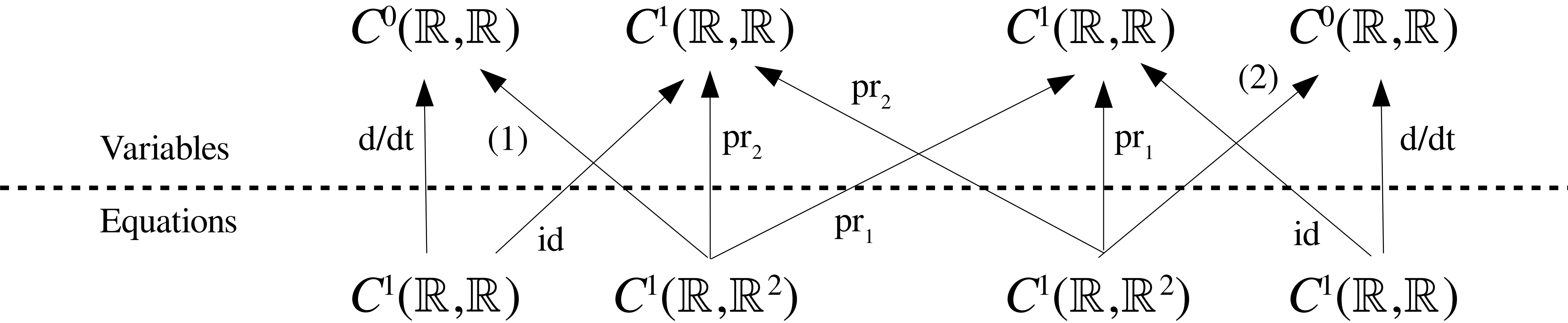
$v$ = Employment rate

$$\dot{u} = \quad u(t)\left(-(\alpha+\gamma) + (\rho v(t))\right). \qquad (2)$$

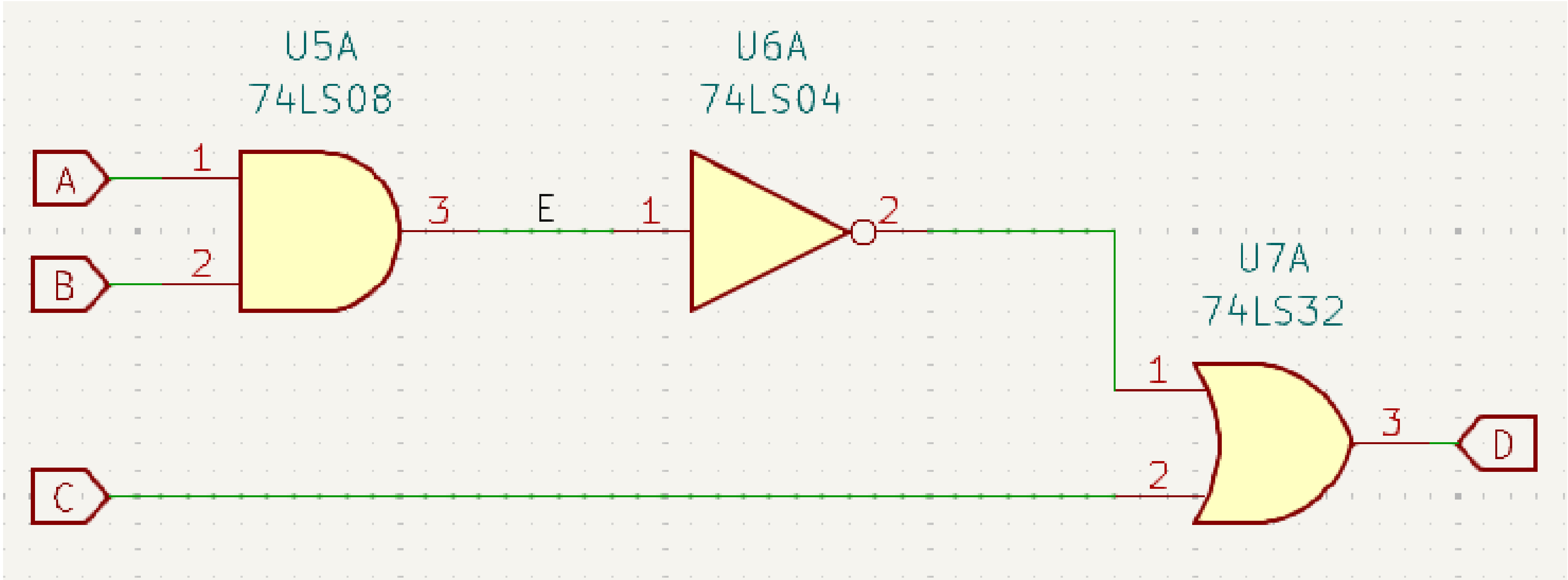
$u$ = Workers' share of income

$$\dot{v} = dv/dt \qquad (3)$$

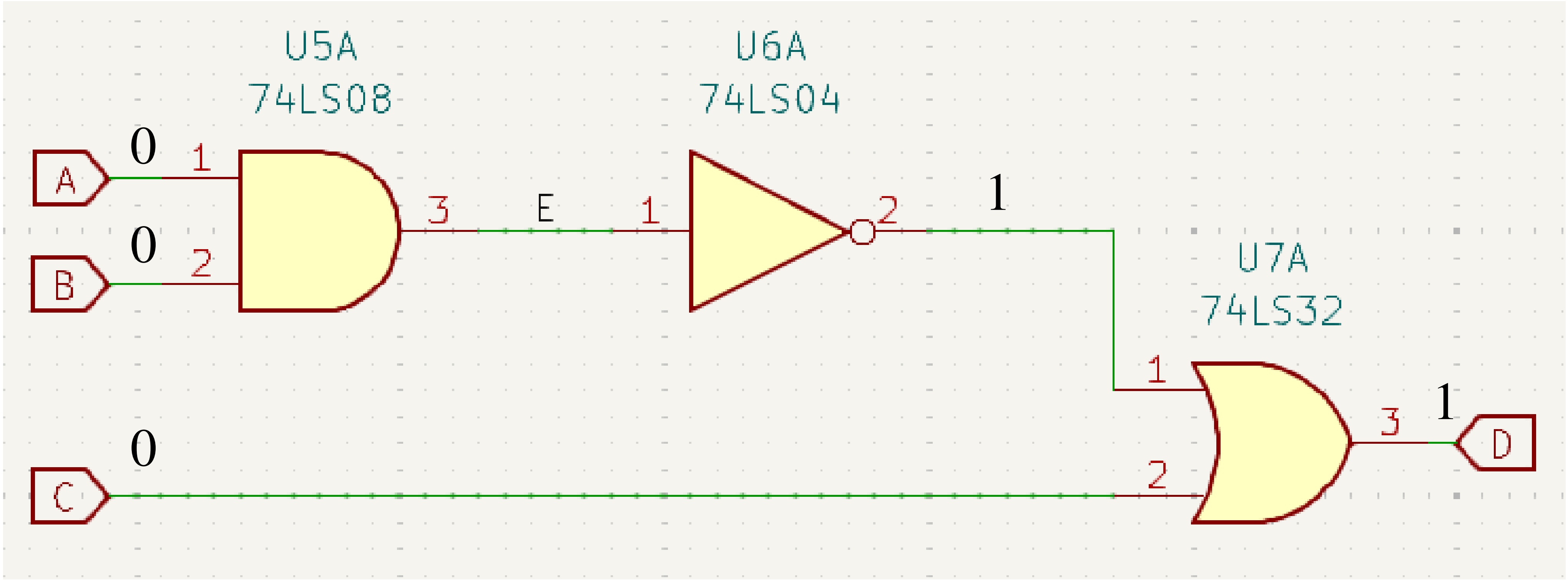
$$\dot{u} = du/dt \qquad (4)$$



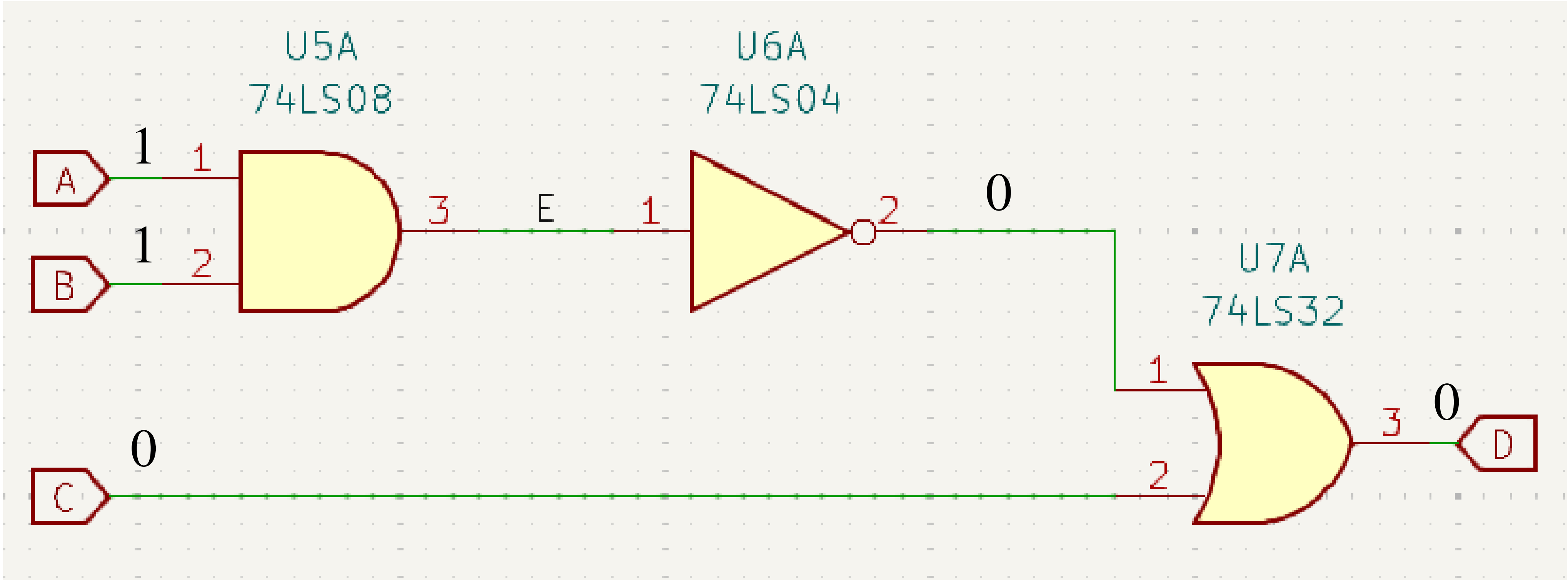
# Also equation systems: Logic circuits



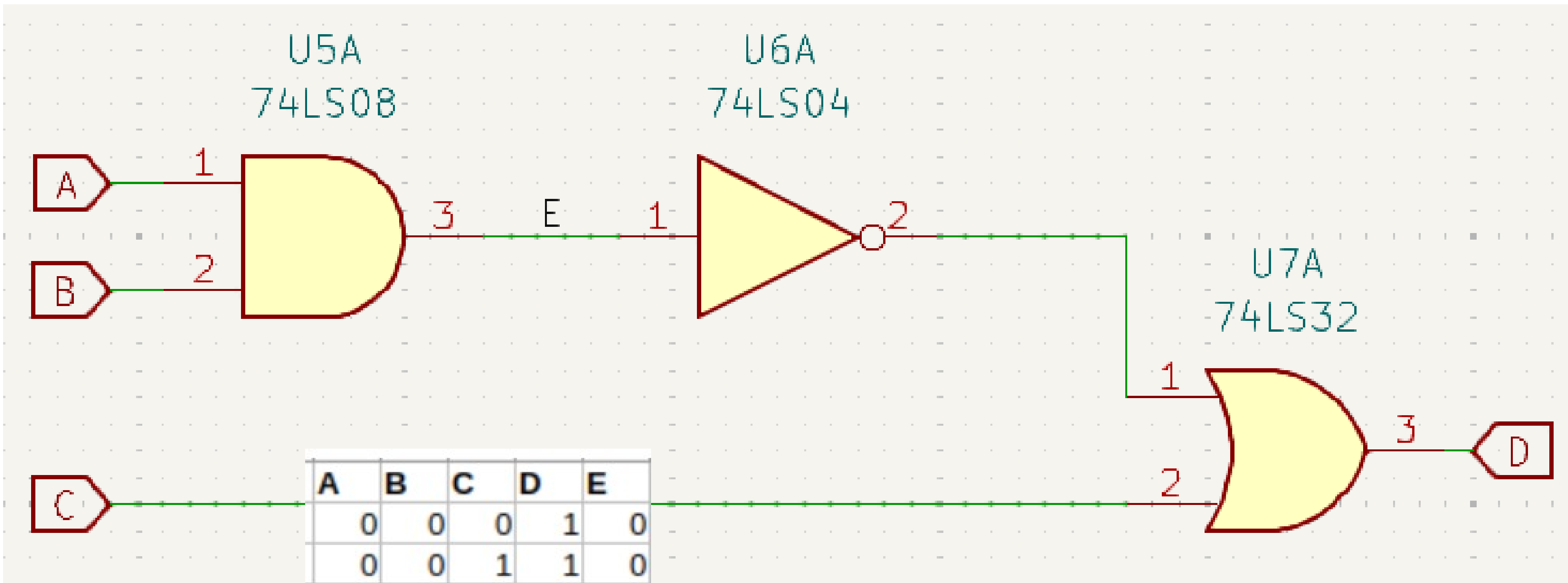
# Logic circuits



# Logic circuits

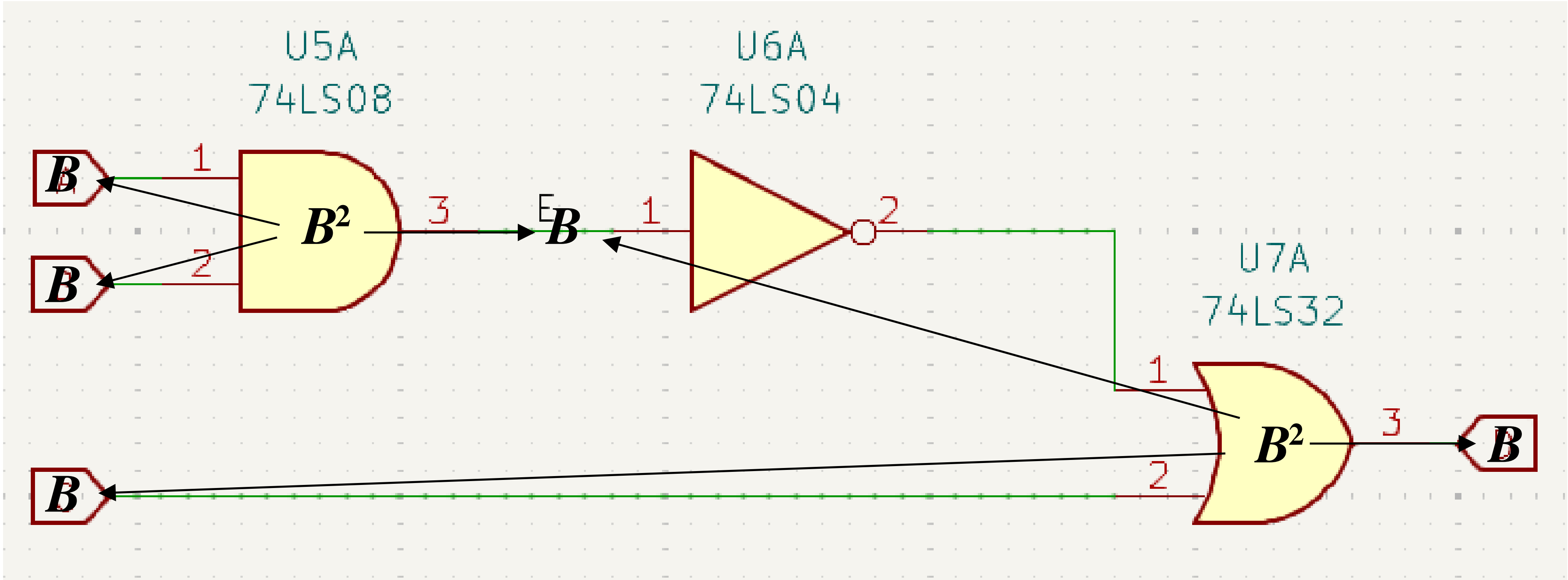


# Logic circuits



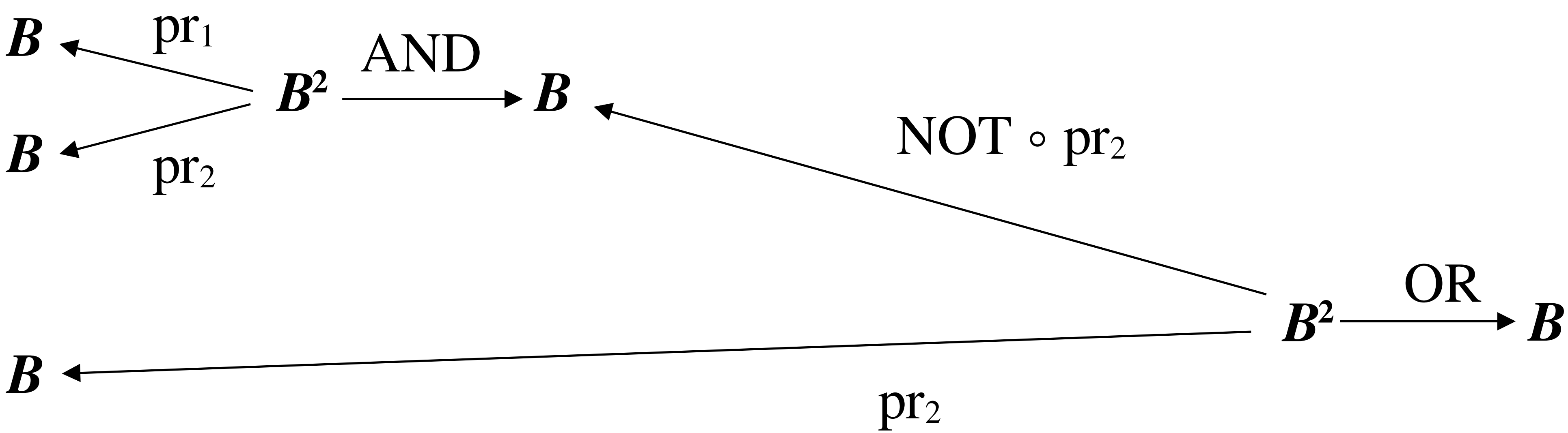

| A | B | C | D | E |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

# Sheafify… via spans!




$\boldsymbol{B} = \{0,1\}$, ie. Boolean values

# Sheafify… via spans!



# Software!

- Logic circuits

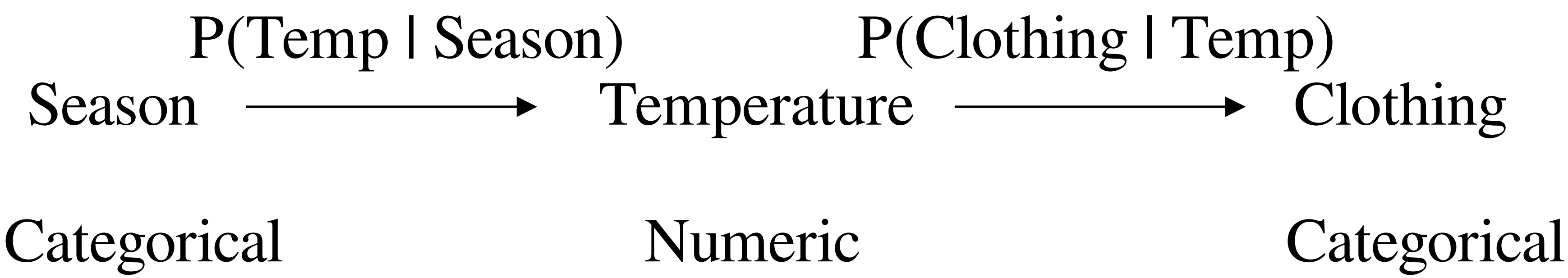
https://colab.research.google.com/drive/1S_c3rQ88JDTTdtBP8VYu5Y7aP7w0F41T

# Bayesian networks (aka "Bayes nets")

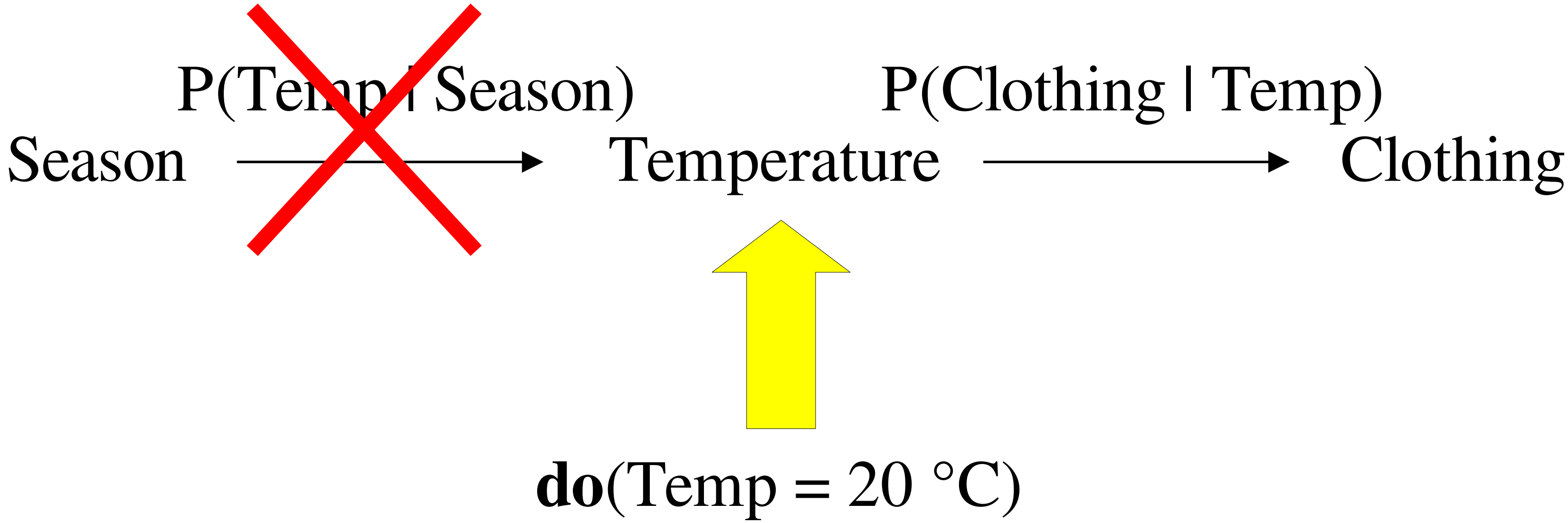
P(Temp | Season) P(Clothing | Temp)

Season → Temperature → Clothing

# Bayesian networks (aka "Bayes nets")



# Causal networks




Let's see what happens when we turn on the A/C;
we don't care about the season any more...

# Let's try to write some equations...

Definition: P($A$ | $B$) P($B$) = P($A$,$B$)

P(Temp | Season) P(Clothing | Temp)

Season ⟶ Temperature ⟶ Clothing

P(Temp,Season) = P(Temp | Season) P(Season)
P(Clothing,Temp) = P(Clothing | Temp) P(Temp)

# Let's try to write some equations...

Definition: $P(A \mid B)\,P(B) = P(A,B)$

P(Temp | Season) P(Clothing | Temp)

Season ⟶ Temperature ⟶ Clothing

P(Temp,Season) = P(Temp | Season) P(Season)
P(Clothing,Temp) = P(Clothing | Temp) P(Temp)

… it seems that we don't have enough equations to fully solve for P(Clothing), say…

… what we're missing are the equations that marginalize out variables from a joint distribution. There are quite a few of these!

# But wait, there's more...

Since there are three variable in play, there are many ways to marginalize all the various joints, including those "not in" the Bayes net

P(Temp | Season) P(Clothing | Temp)

Season → Temperature → Clothing

$$P(\text{Temp},\text{Season}) = P(\text{Temp} \mid \text{Season})\, P(\text{Season})$$
$$P(\text{Clothing},\text{Temp}) = P(\text{Clothing} \mid \text{Temp})\, P(\text{Temp})$$
$$P(\text{Season}) = \Sigma_t\, P(\text{Temp} = t, \text{Season})$$
$$P(\text{Temp}) = \Sigma_s\, P(\text{Temp}, \text{Season} = s)$$
$$P(\text{Temp}) = \Sigma_c\, P(\text{Clothing} = c, \text{Temp})$$
$$P(\text{Clothing}) = \Sigma_t\, P(\text{Clothing}, \text{Temp} = t)$$
$$P(\text{Season}) = \Sigma_c\, P(\text{Clothing} = c, \text{Season})$$
$$P(\text{Clothing}) = \Sigma_s\, P(\text{Clothing}, \text{Season} = s)$$

# But wait, there's **even** more...

We forgot the three-way marginals too!  (But this is now everything)

$$\text{Season} \xrightarrow{P(\text{Temp} \mid \text{Season})} \text{Temperature} \xrightarrow{P(\text{Clothing} \mid \text{Temp})} \text{Clothing}$$

$P(\text{Temp},\text{Season}) = P(\text{Temp} \mid \text{Season})\, P(\text{Season})$

$P(\text{Clothing},\text{Temp}) = P(\text{Clothing} \mid \text{Temp})\, P(\text{Temp})$

$P(\text{Season}) = \Sigma_t\, P(\text{Temp} = t, \text{Season})$

$P(\text{Temp}) = \Sigma_s\, P(\text{Temp}, \text{Season} = s)$

$P(\text{Temp}) = \Sigma_c\, P(\text{Clothing} = c, \text{Temp})$

$P(\text{Clothing}) = \Sigma_t\, P(\text{Clothing}, \text{Temp} = t)$

$P(\text{Season}) = \Sigma_c\, P(\text{Clothing} = c, \text{Season})$

$P(\text{Clothing}) = \Sigma_s\, P(\text{Clothing}, \text{Season} = s)$

$P(\text{Season},\text{Temp}) = \Sigma_c\, P(\text{Clothing} = c, \text{Temp}, \text{Season})$

$P(\text{Season},\text{Clothing}) = \Sigma_t\, P(\text{Clothing}, \text{Temp} = t, \text{Season})$

$P(\text{Temp},\text{Clothing}) = \Sigma_s\, P(\text{Clothing}, \text{Temp}, \text{Season} = s)$

# We'd like to sheafify...

… but things are getting very busy; let's summarize the names

$$X_1 \xrightarrow{P(X_2 \mid X_1)} X_2 \xrightarrow{P(X_3 \mid X_2)} X_3$$

$P(X_1,X_2) = P(X_2 \mid X_1)\, P(X_1)$
$P(X_2,X_3) = P(X_3 \mid X_2)\, P(X_2)$

2 conditional equations

$P(X_1) = \Sigma_t\, P(X_2 = t, X_1)$
$P(X_2) = \Sigma_s\, P(X_2, X_1 = s)$
$P(X_2) = \Sigma_c\, P(X_3 = c, X_2)$
$P(X_3) = \Sigma_t\, P(X_3, X_2 = t)$
$P(X_1) = \Sigma_c\, P(X_3 = c, X_1)$
$P(X_3) = \Sigma_s\, P(X_3, X_1 = s)$
$P(X_1,X_2) = \Sigma_c\, P(X_3 = c, X_2, X_1)$
$P(X_1,X_3) = \Sigma_t\, P(X_3, X_2 = t, X_1)$
$P(X_2,\mathrm{X}_3) = \Sigma_s\, P(X_3, X_2, X_1 = s)$

9 Marginal equations

# What are the stalks & restrictions?

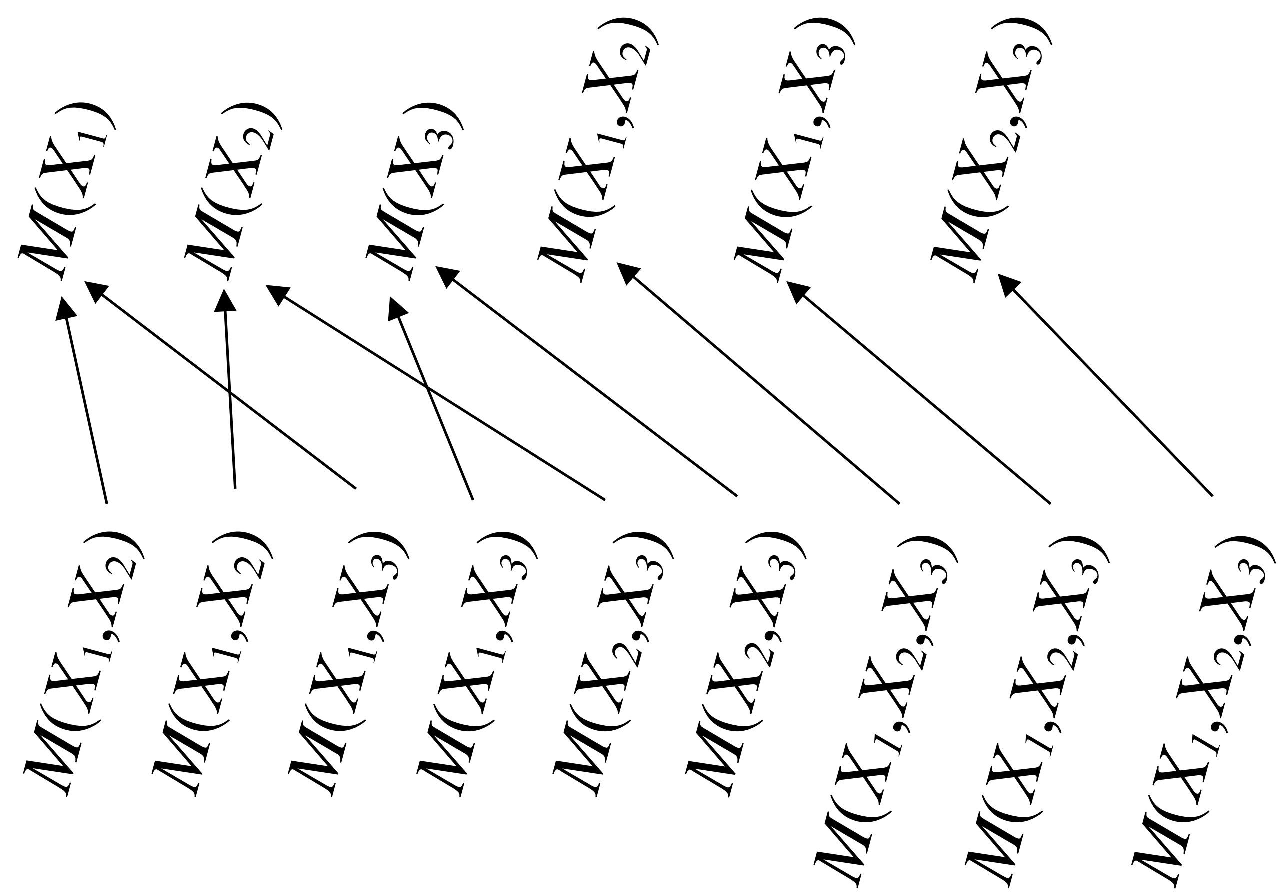
Each "variable" in our system of equations is a probability distribution

<u>Definition</u>: $M(X_1,X_2,X_3)$ is the set of joint probability distributions on $X_1,X_2,X_3$. (Similar for more/fewer variables)

| Equations: | Restriction types: |
|---|---|
| $P(X_1,X_2) = P(X_2 \mid X_1)\, P(X_1)$ | $M(X_1) \to M(X_1,X_2)$ |
| $P(X_2,X_3) = P(X_3 \mid X_2)\, P(X_2)$ | $M(X_2) \to M(X_2,X_3)$ |
| $P(X_1) = \Sigma_t\, P(X_2 = t, X_1)$ | $M(X_1,X_2) \to M(X_1)$ |
| $P(X_2) = \Sigma_s\, P(X_2, X_1 = s)$ | $M(X_1,X_2) \to M(X_2)$ |
| $P(X_2) = \Sigma_c\, P(X_3 = c, X_2)$ | $M(X_2,X_3) \to M(X_2)$ |
| $P(X_3) = \Sigma_t\, P(X_3, X_2 = t)$ | $M(X_2,X_3) \to M(X_3)$ |
| $P(X_1) = \Sigma_c\, P(X_3 = c, X_1)$ | $M(X_1,X_3) \to M(X_1)$ |
| $P(X_3) = \Sigma_s\, P(X_3, X_1 = s)$ | $M(X_1,X_3) \to M(X_3)$ |
| $P(X_1,X_2) = \Sigma_c\, P(X_3 = c, X_2, X_1)$ | $M(X_1,X_2,X_3) \to M(X_1,X_2)$ |
| $P(X_1,X_3) = \Sigma_t\, P(X_3, X_2 = t, X_1)$ | $M(X_1,X_2,X_3) \to M(X_1,X_3)$ |
| $P(X_2,X_3) = \Sigma_s\, P(X_3, X_2, X_1 = s)$ | $M(X_1,X_2,X_3) \to M(X_2,X_3)$ |

# Bayesian network as a sheaf

- Marginals… (Always present)

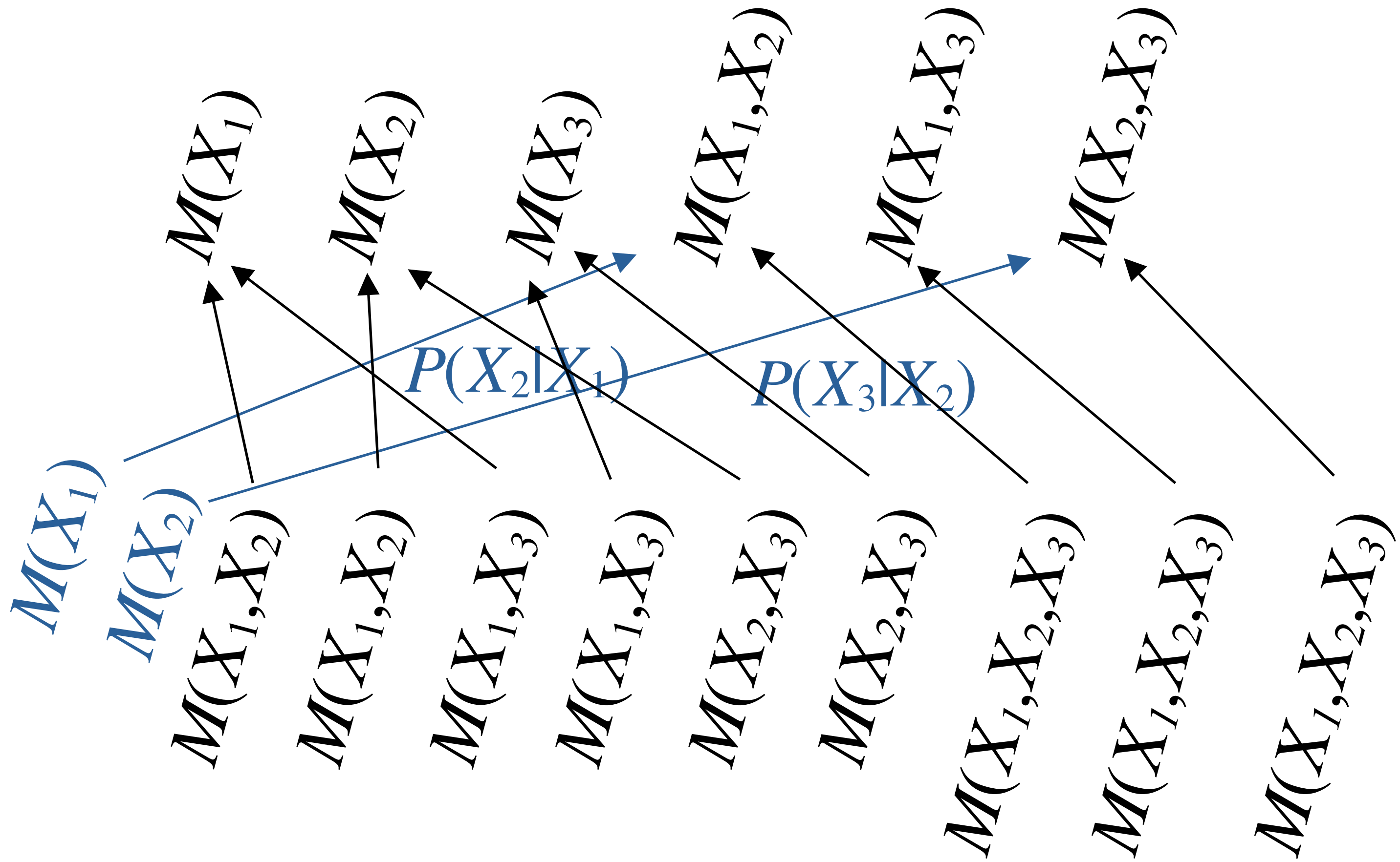


# Bayesian network as a sheaf

- … conditionals … (based upon the Bayes net)

$X_1 \rightarrow X_2 \rightarrow X_3$

Bayes net

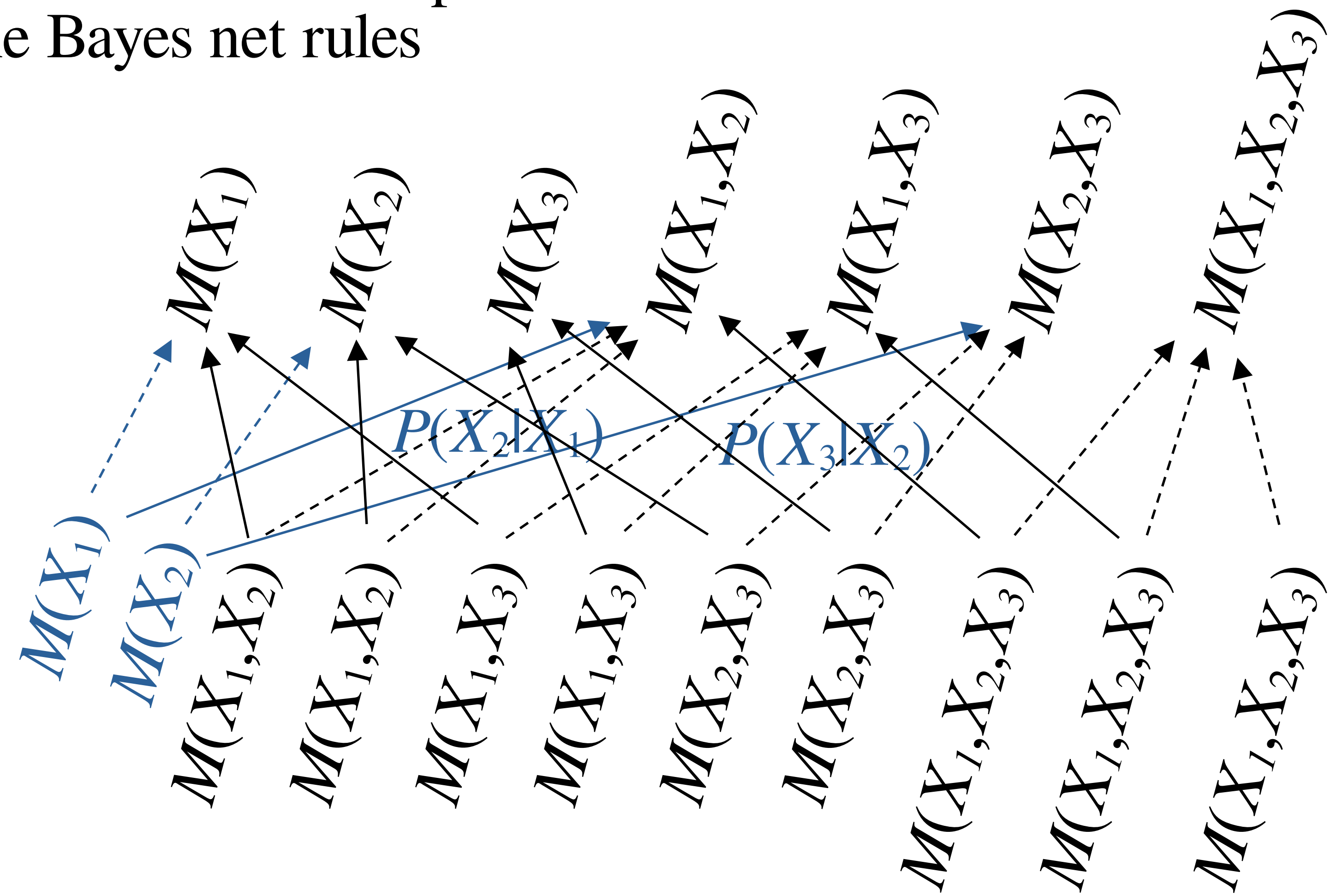


# Bayesian network as a sheaf

- … identities! (Added to ensure consistency across copies.)
- <u>Corollary</u>: Global sections are possible sets of distributions that satisfy the Bayes net rules

$X_1 \rightarrow X_2 \rightarrow X_3$

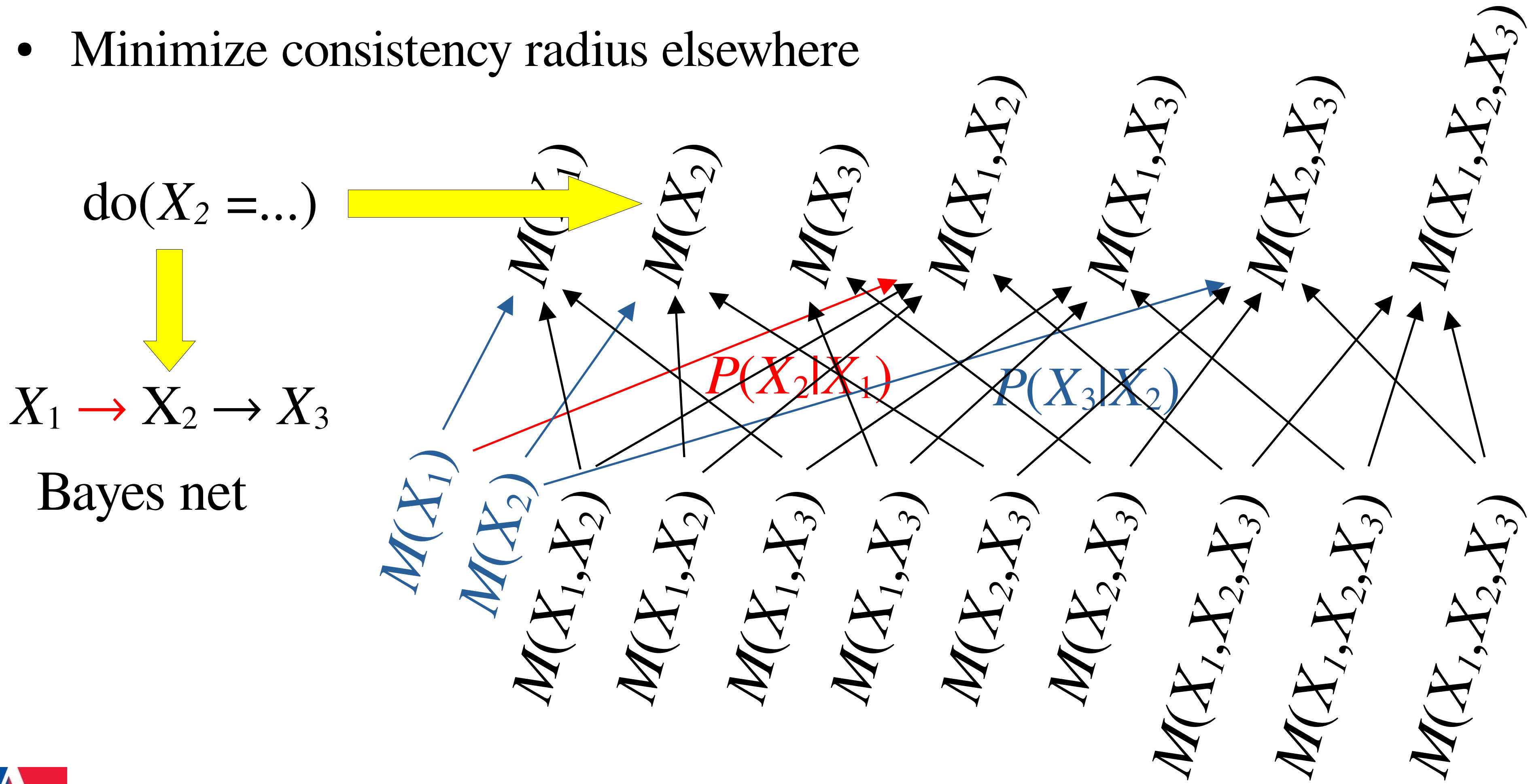
Bayes net




Olivia Chen Michael Robinson

# Causal modeling using **do** operator

- Make an assignment to variable in top row with P(desired value) = 1
- Delete the **conditional arrows** (leave the marginals) into that variable
- Minimize consistency radius elsewhere



# Consistency: Discretizing correctly

# Discretization of functions

$$C^k(X,Y) \longrightarrow \mathbb{R}^n$$

$$f \qquad\qquad (f(x_1),\ldots,f(x_n))$$

# Discretization of functions

$$\mathbb{R}^m \longrightarrow C^k(X,Y) \longrightarrow \mathbb{R}^n$$

$$(a_1,\ldots,a_m) \qquad f = \sum a_i f_i(x)$$

# Why discretize?

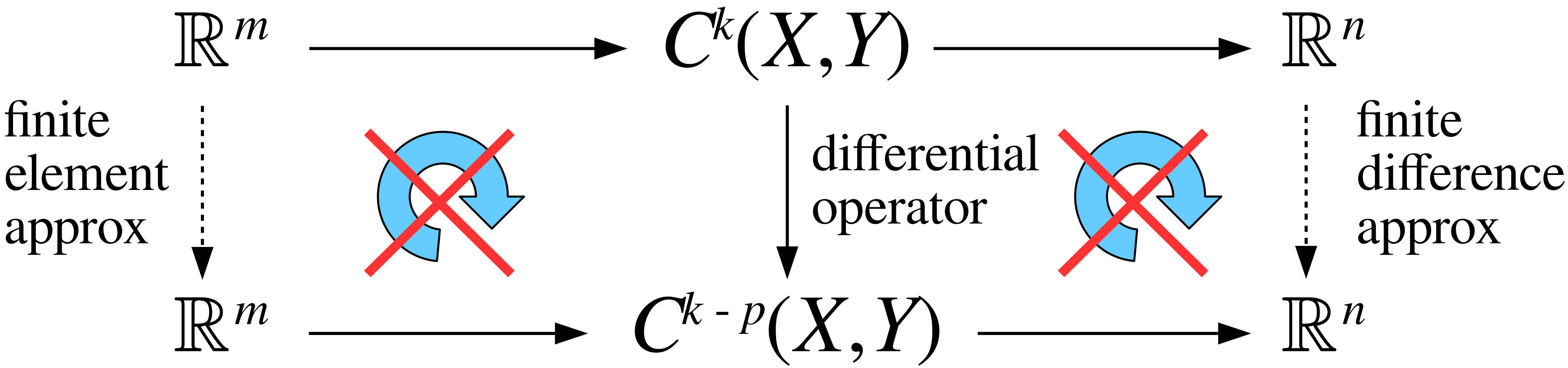
$$\mathbb{R}^m \longrightarrow C^k(X,Y) \longrightarrow \mathbb{R}^n$$

$$C^k(X,Y) \xrightarrow{\text{differential operator}} C^{k-p}(X,Y)$$

# Why discretize?

$$
\begin{array}{ccccc}
\mathbb{R}^m & \longrightarrow & C^k(X,Y) & \longrightarrow & \mathbb{R}^n \\
 & & \Big\downarrow \text{differential operator} & & \\
\mathbb{R}^m & \longrightarrow & C^{k-p}(X,Y) & \longrightarrow & \mathbb{R}^n
\end{array}
$$

# Why discretize?

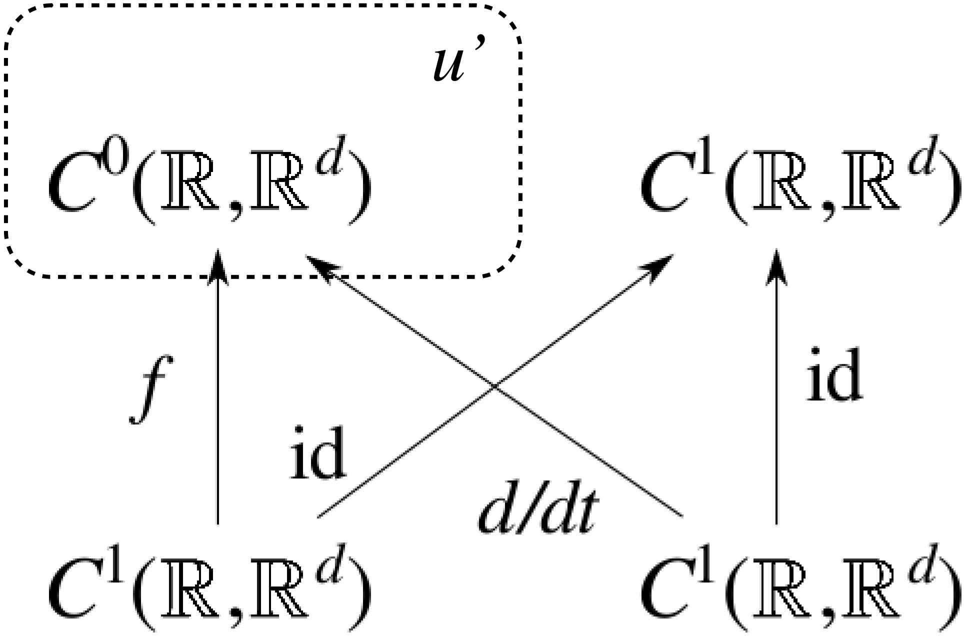



Goals:

1. Make the diagram commute as $m$, $n \to \infty$ (*consistency* of the approximation)
2. Recover properties of the differential operator from the approximations (*convergence* of the approximation)

# Back to our original example

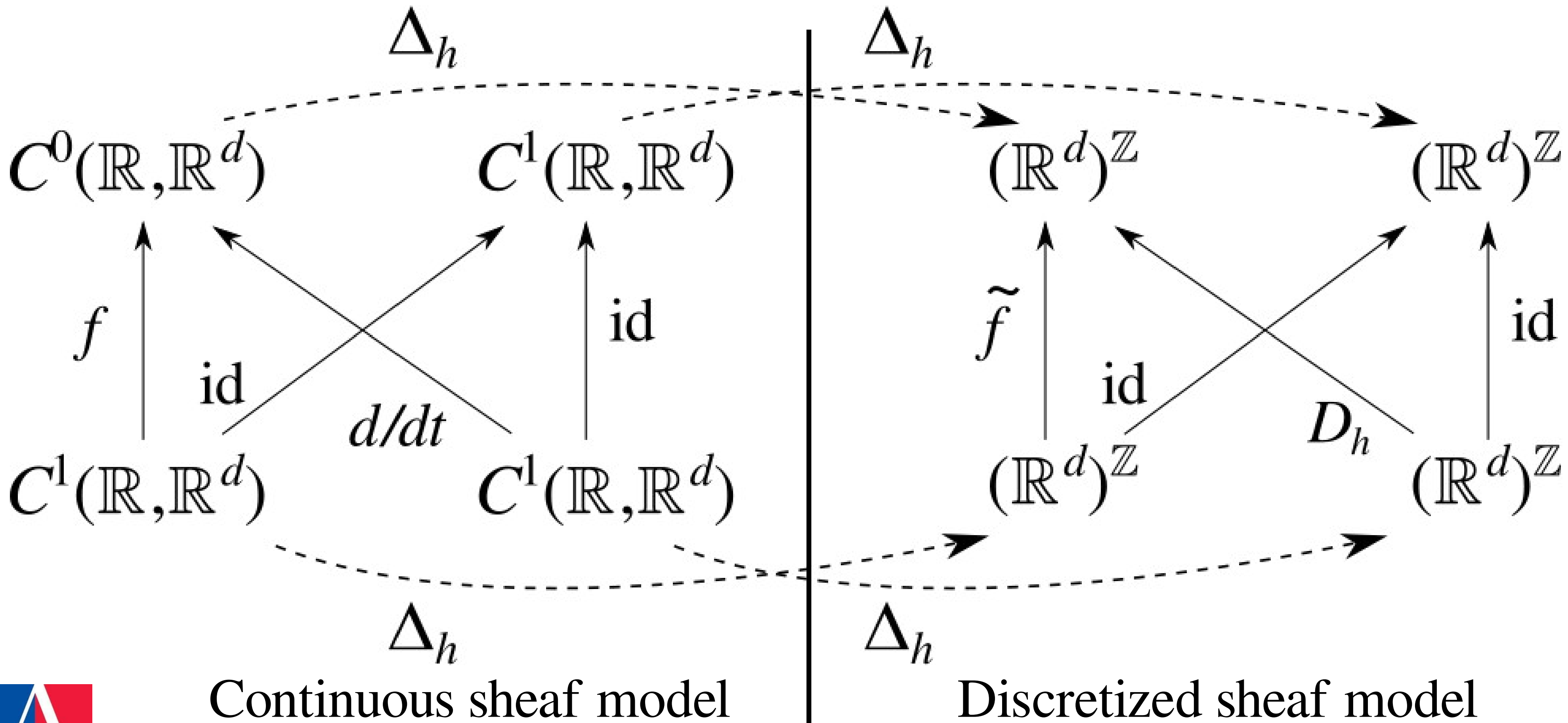
- Consider $u' = f(u)$ on the real line
- This has a sheaf diagram



# Finite differences

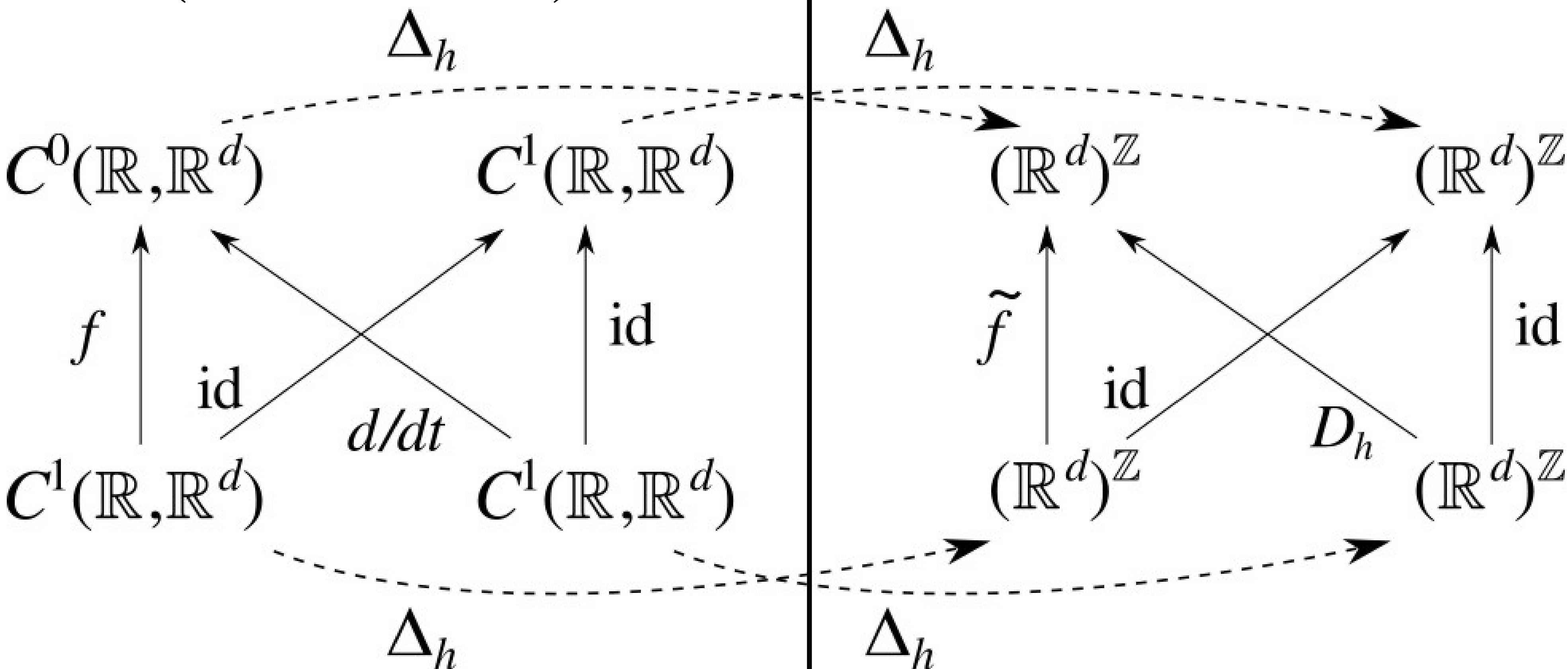
- Discretizing each function space via a fixed step $h$

$$(\Delta_h u)_n = u(nh)$$



# Is it a *sheaf morphism*?

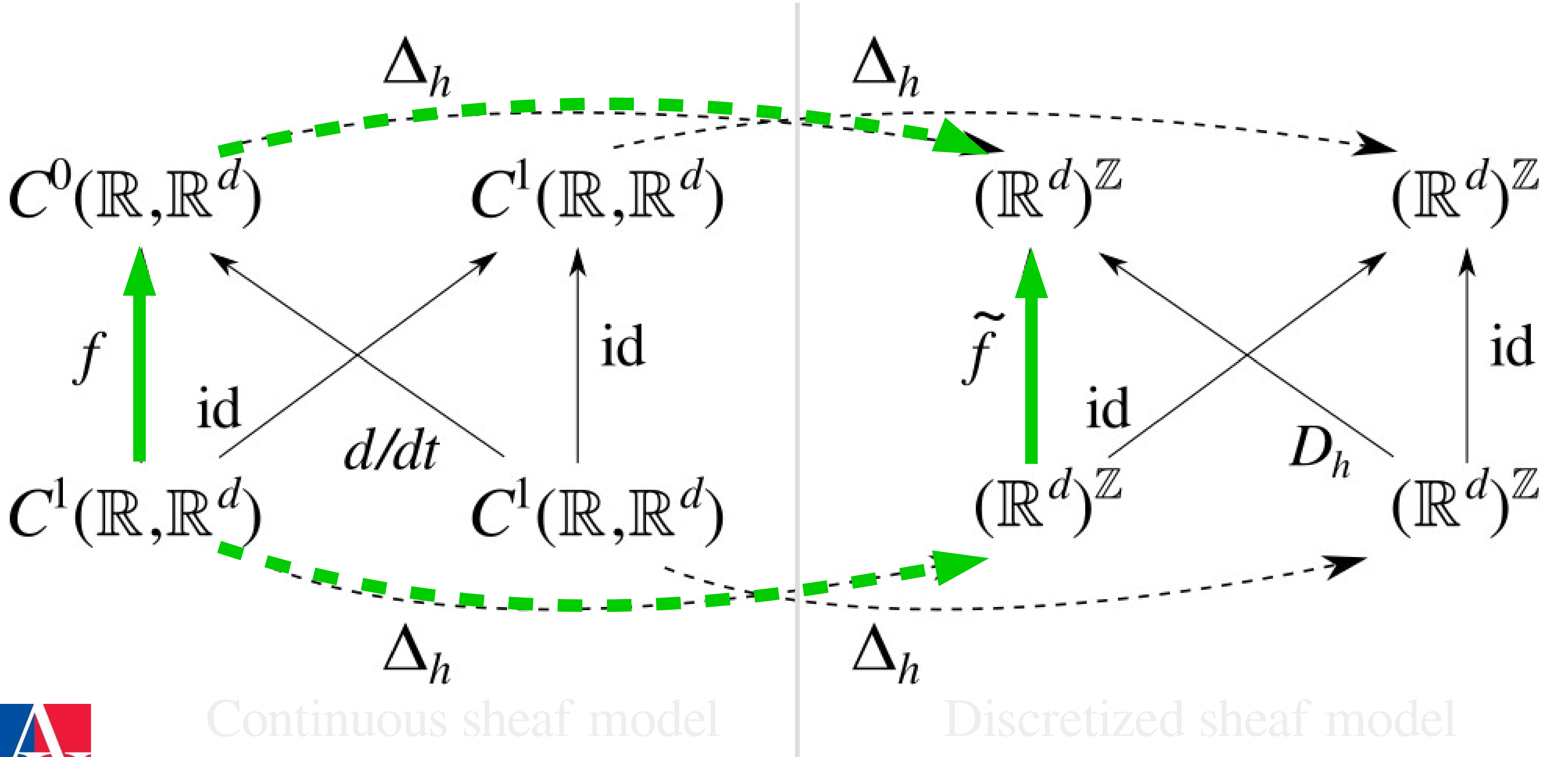
- A *sheaf morphism* is a commutative diagram of maps between stalks of two sheaves… is this one? (dotted arrows)




Continuous sheaf model | Discretized sheaf model

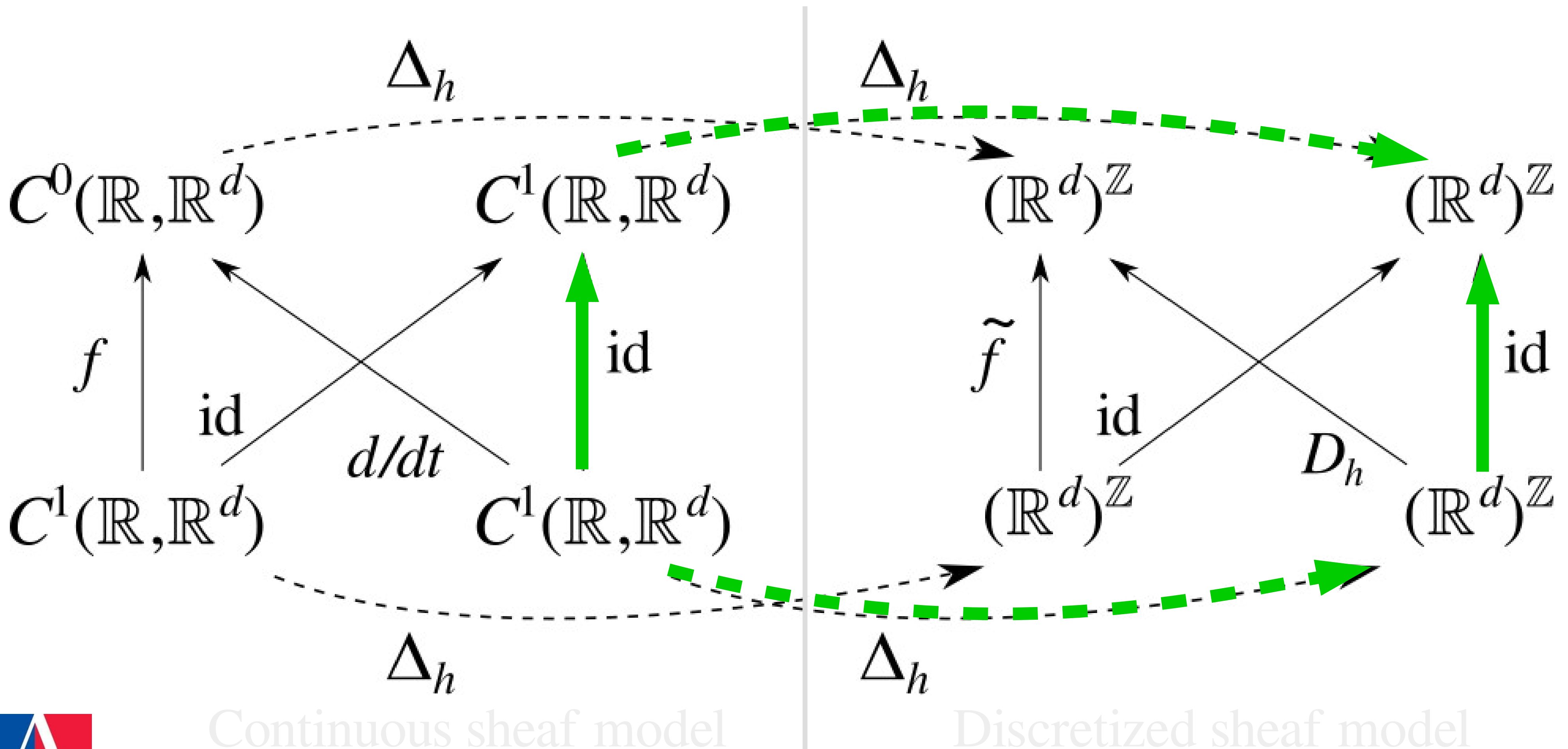
# Is it a *sheaf morphism*?

- This square commutes if we pick $\tilde{f}$ correctly...



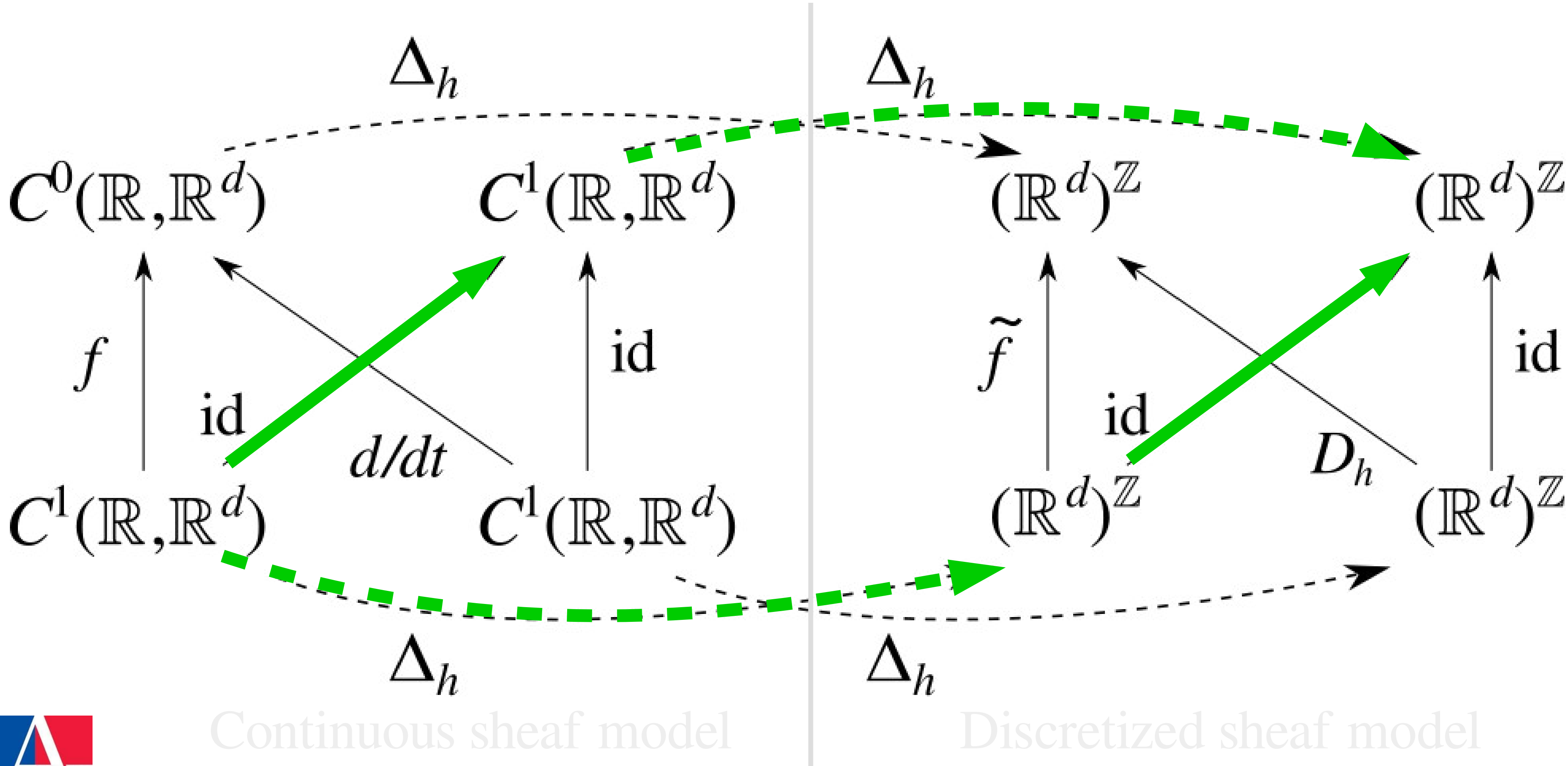
# Is it a *sheaf morphism*?

- … this one commutes trivially …



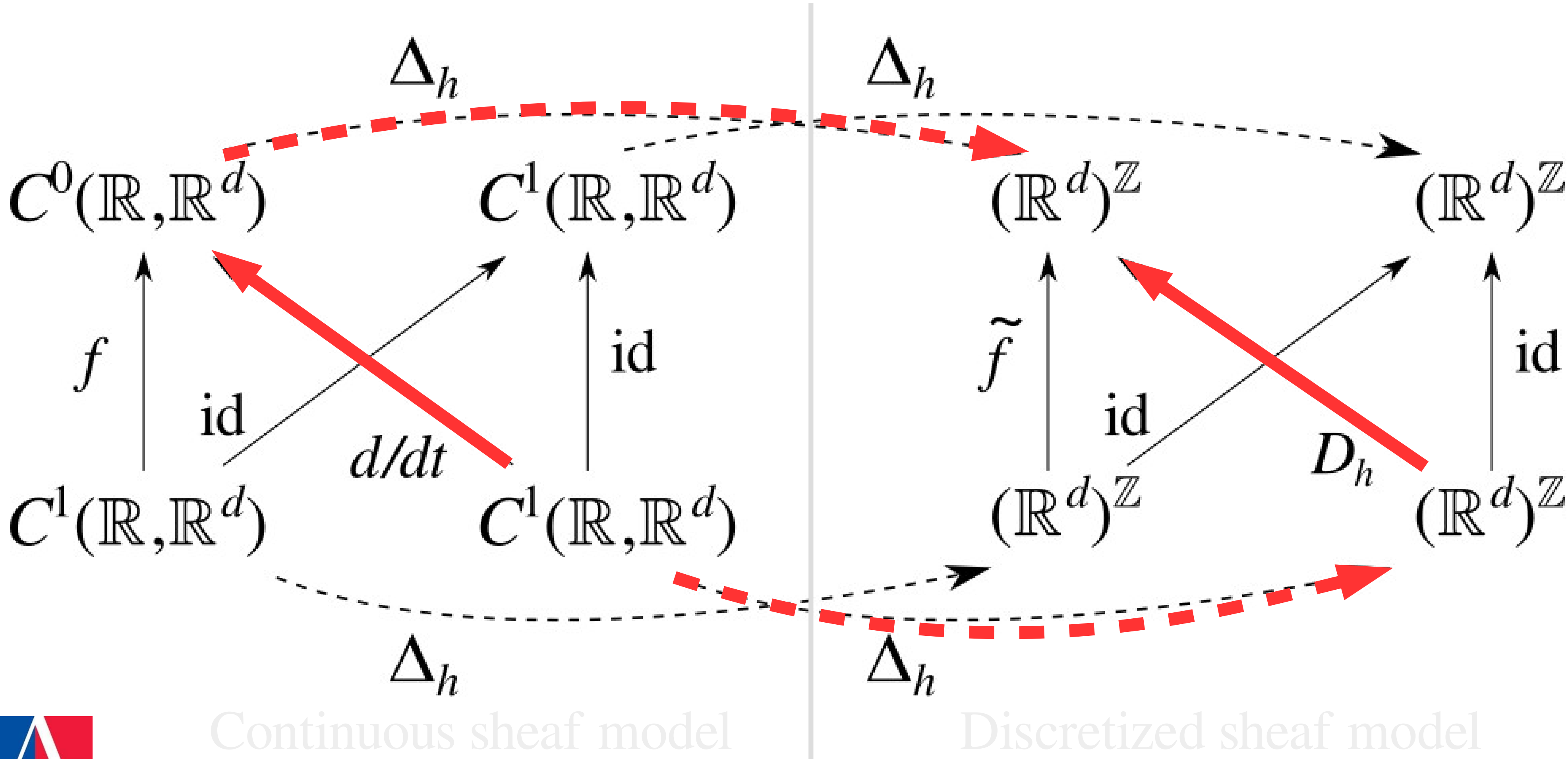
# Is it a *sheaf morphism*?

- … this one also commutes trivially …



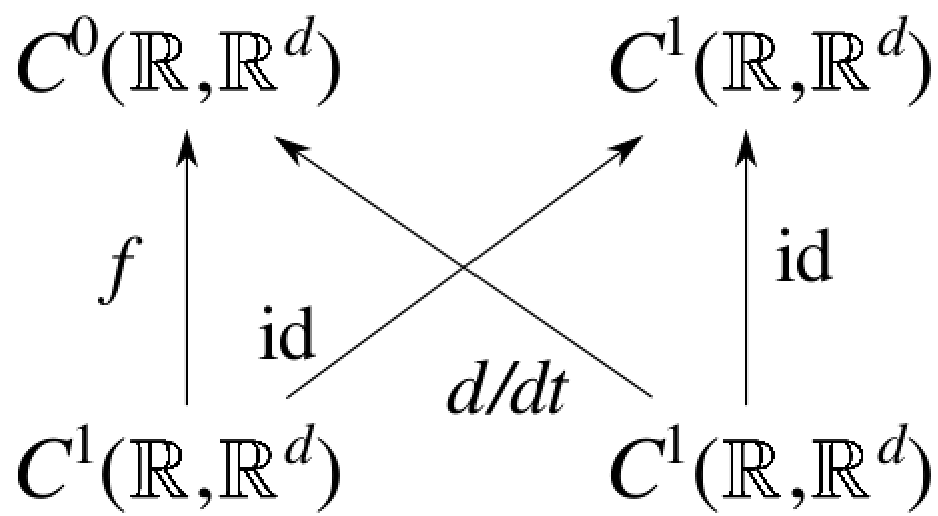
# Is it a *sheaf morphism*?

- ...but this asks that $u'(nh) = D_h u_n$, which means discretized version is **exactly correct**. Oops!



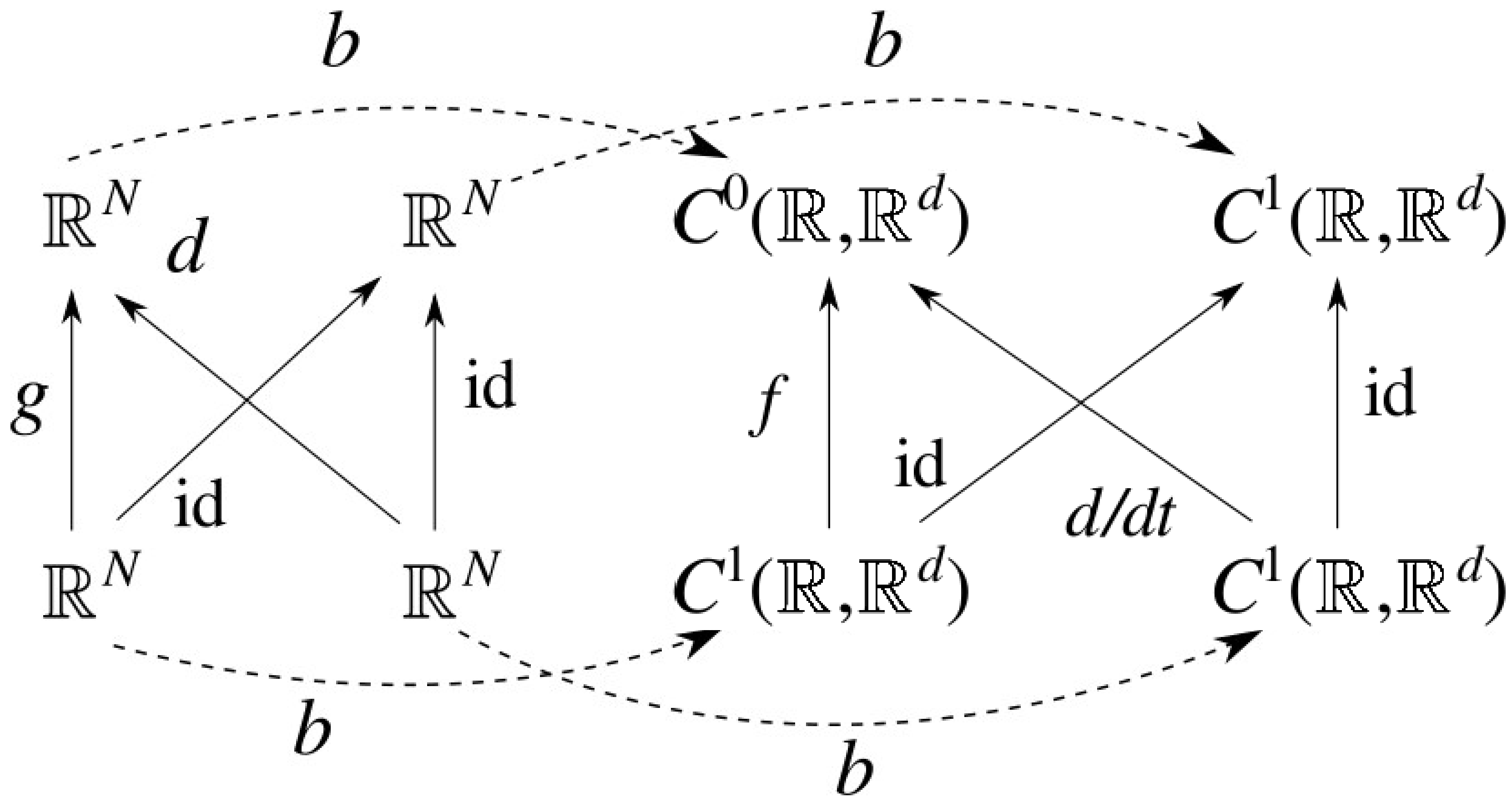
# Finite elements

- We can also try to construct a finite elements approximation… from the “other side”
- Again start with the same continuous sheaf model



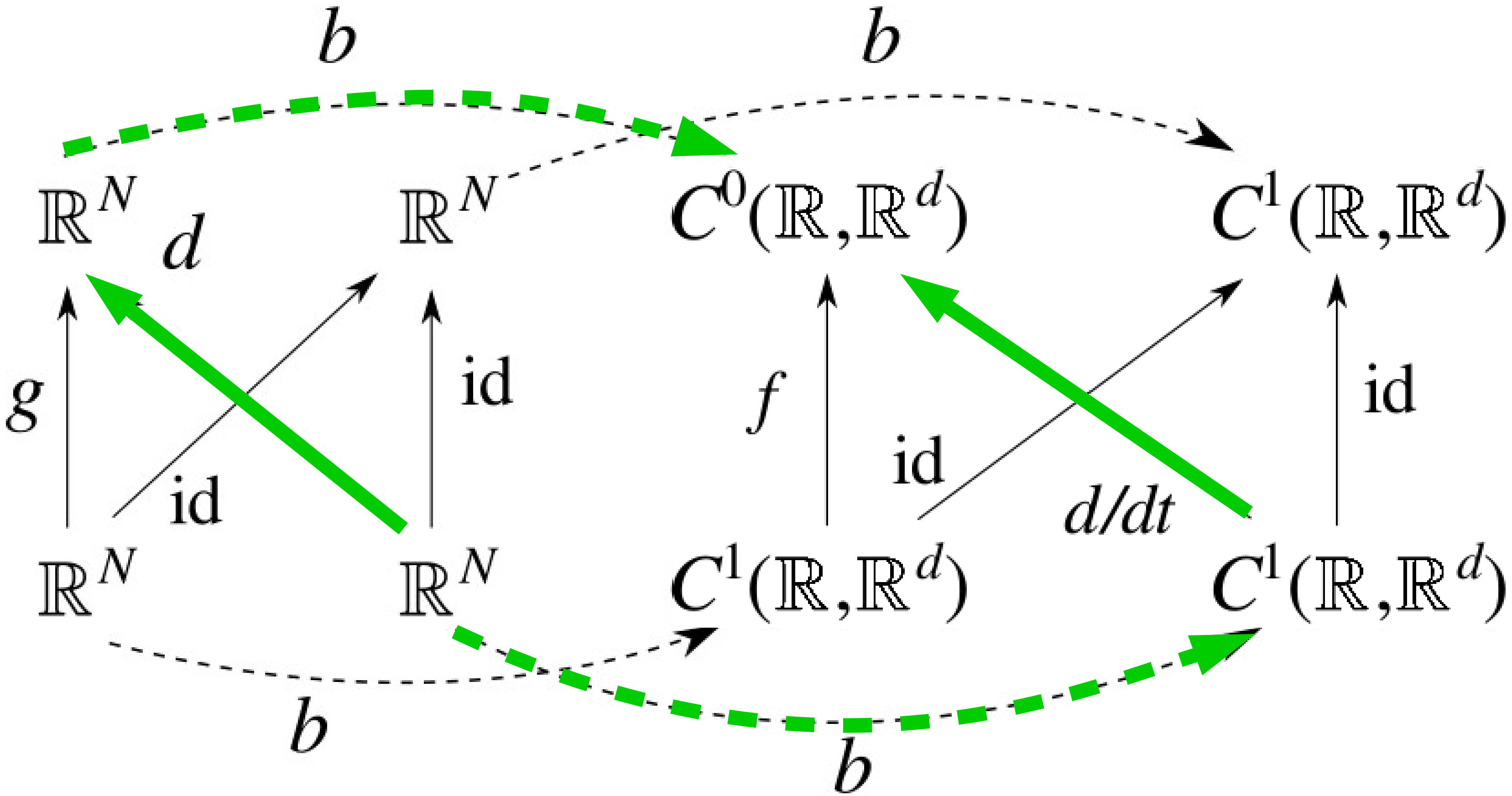
# Finite elements sheaf model

- Use an $N$ dimensional subspace of functions with a linear embedding $b : \mathbb{R}^N \to B \subseteq C^1(\mathbb{R}, \mathbb{R}^d)$.



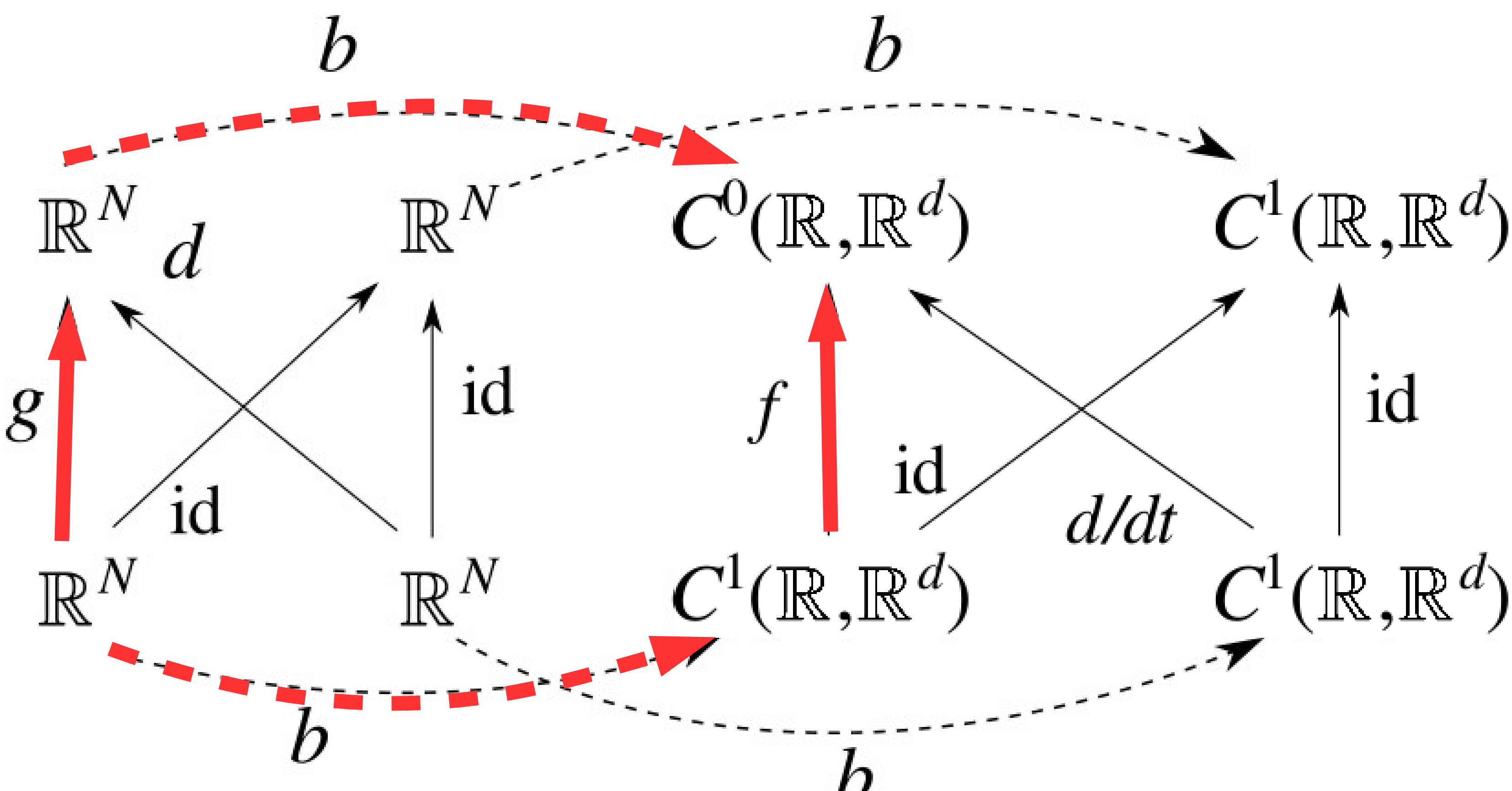
# Is it a sheaf morphism?

- Although the derivative approximation can now be corrected by a judicious choice of embedding $b$…

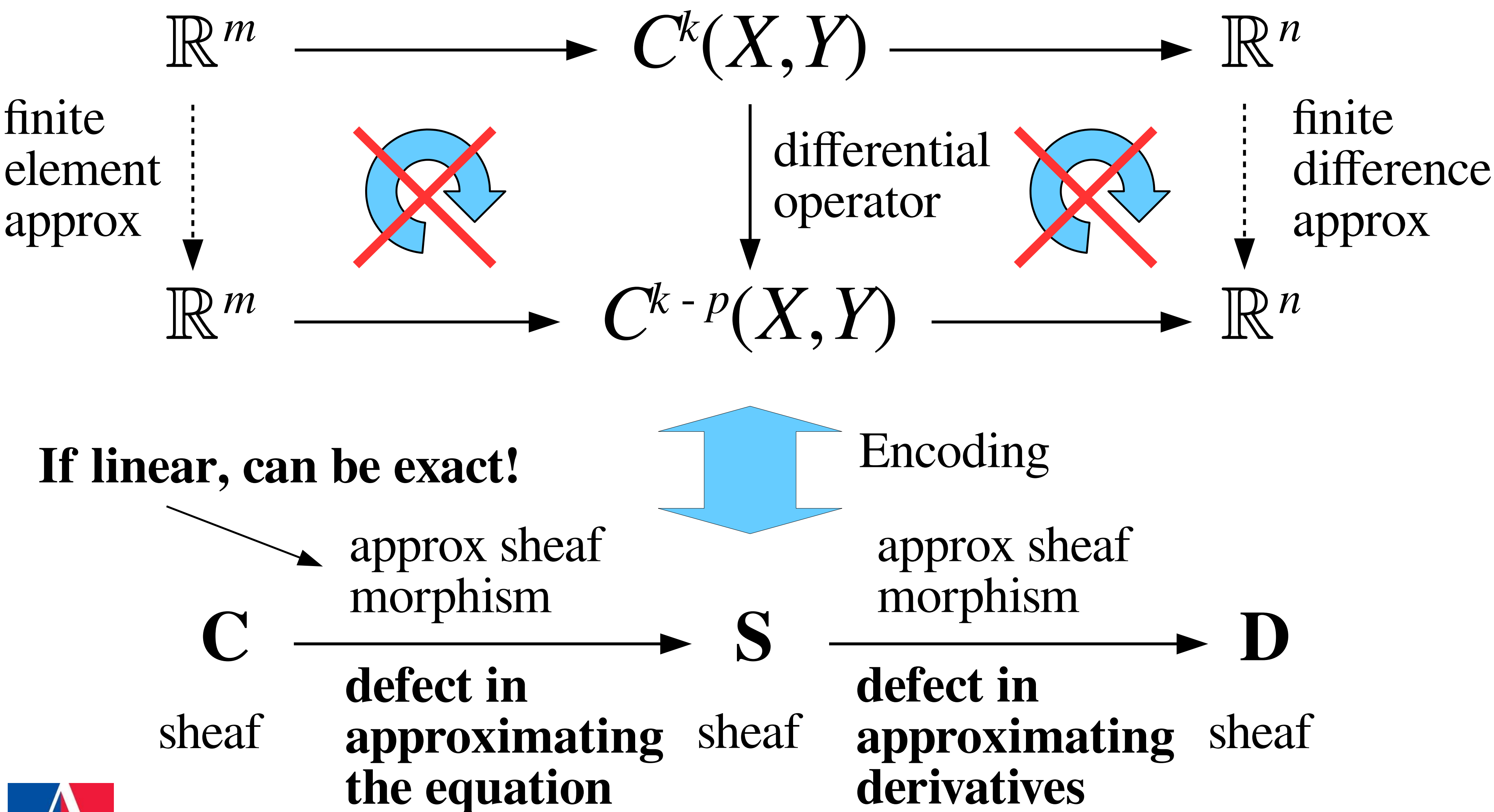


# Might be a sheaf morphism...

- …if not linear, now the equation itself fails
- …if linear, we may get a morphism; *Galerkin method*.



# Observations about consistency



# In summary...

Sheaves capture variable relationships in any system of equations; that's most scientific models!

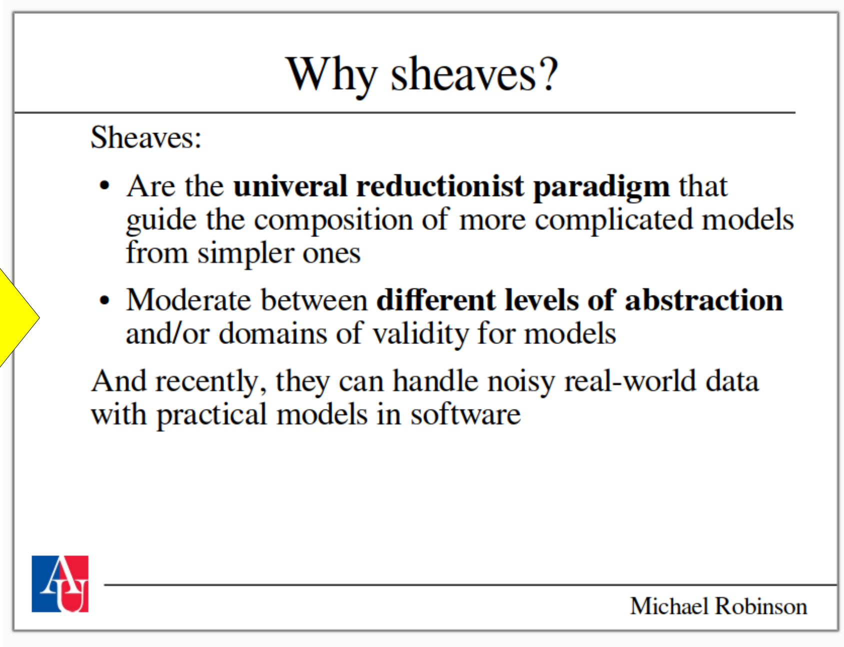
- Differential equation systems
- Bayes & causal nets
- … basically anything described by equations

Consistency radius estimates:

- Measurement error,
- Data modeling error, and
- Discretization error

# We've now seen...

- Building of several sheaf models
- Inferring/imputing missing or noisy data using the sheaf
- But what of the domain of validity?




Reference: https://doi.org/10.32408/compositionality-2-2

# What's the right domain of validity?

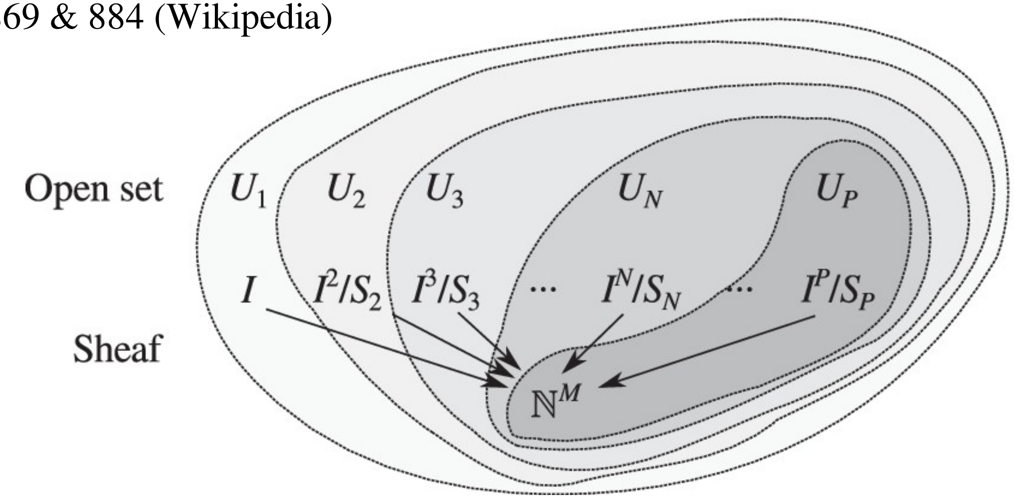
- How many variables do you really need?
- Concrete example: counting stars in a star cluster

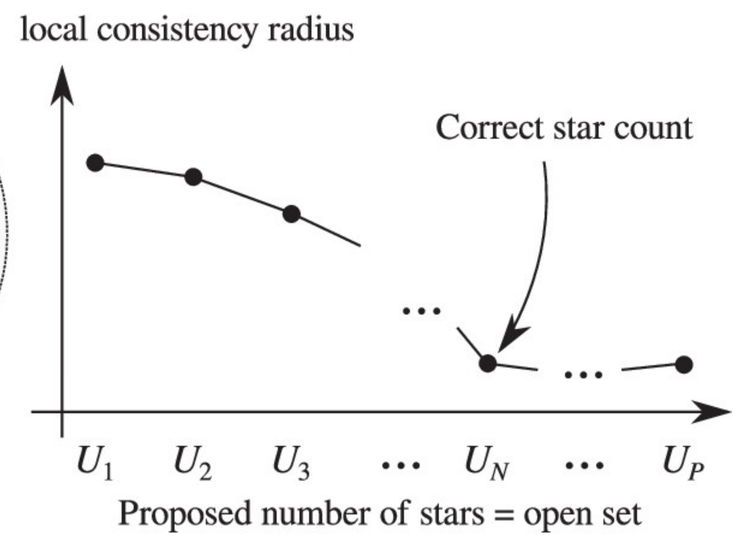
NGC 869 & 884 (Wikipedia)

$I$ = Position and brightness

Variables for $P$ stars: $I^P/S_P$

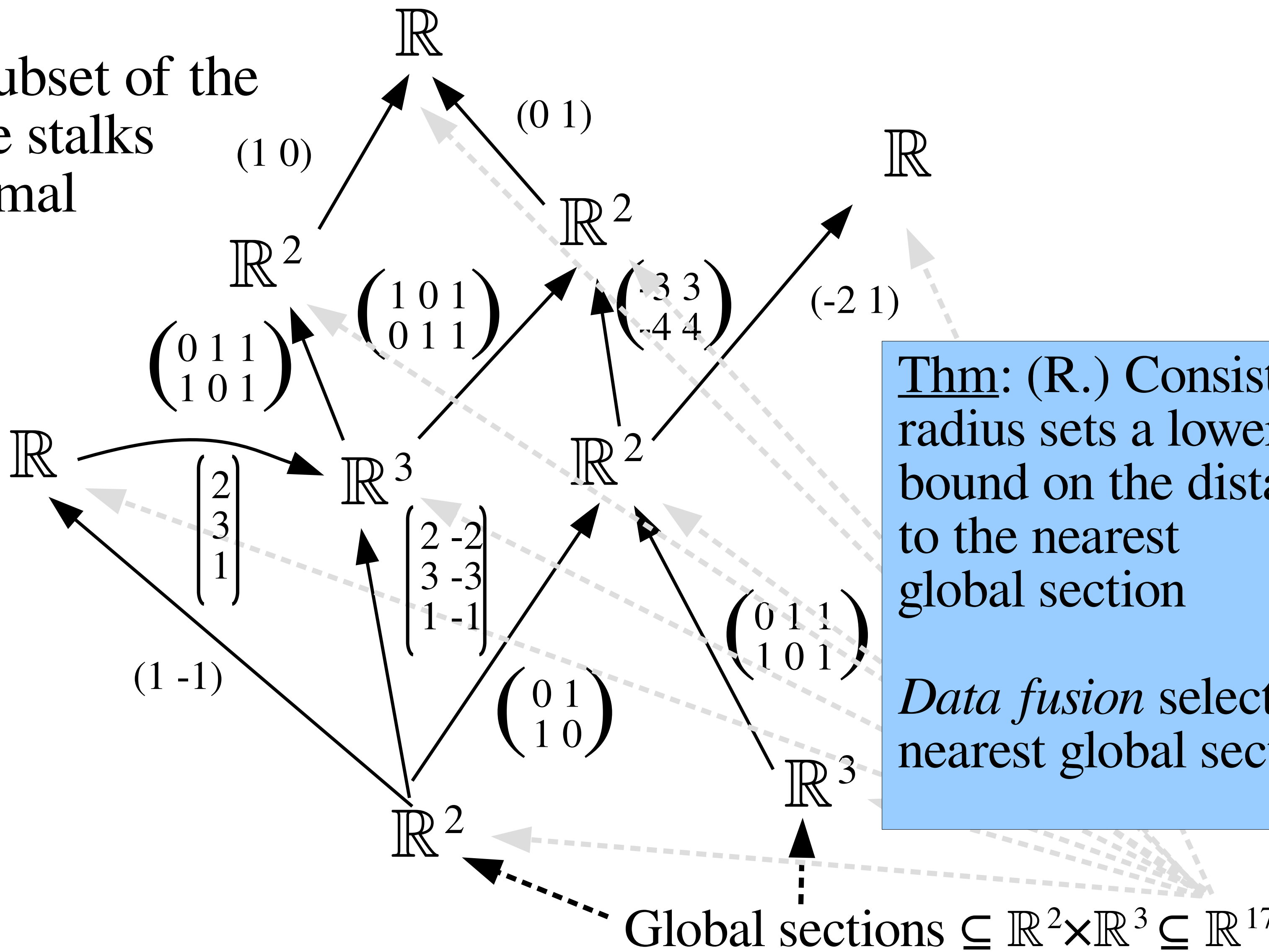
"Symmetric group on P elements" = ignore order



# The space of global sections

It's a closed subset of the product of the stalks over the minimal elements

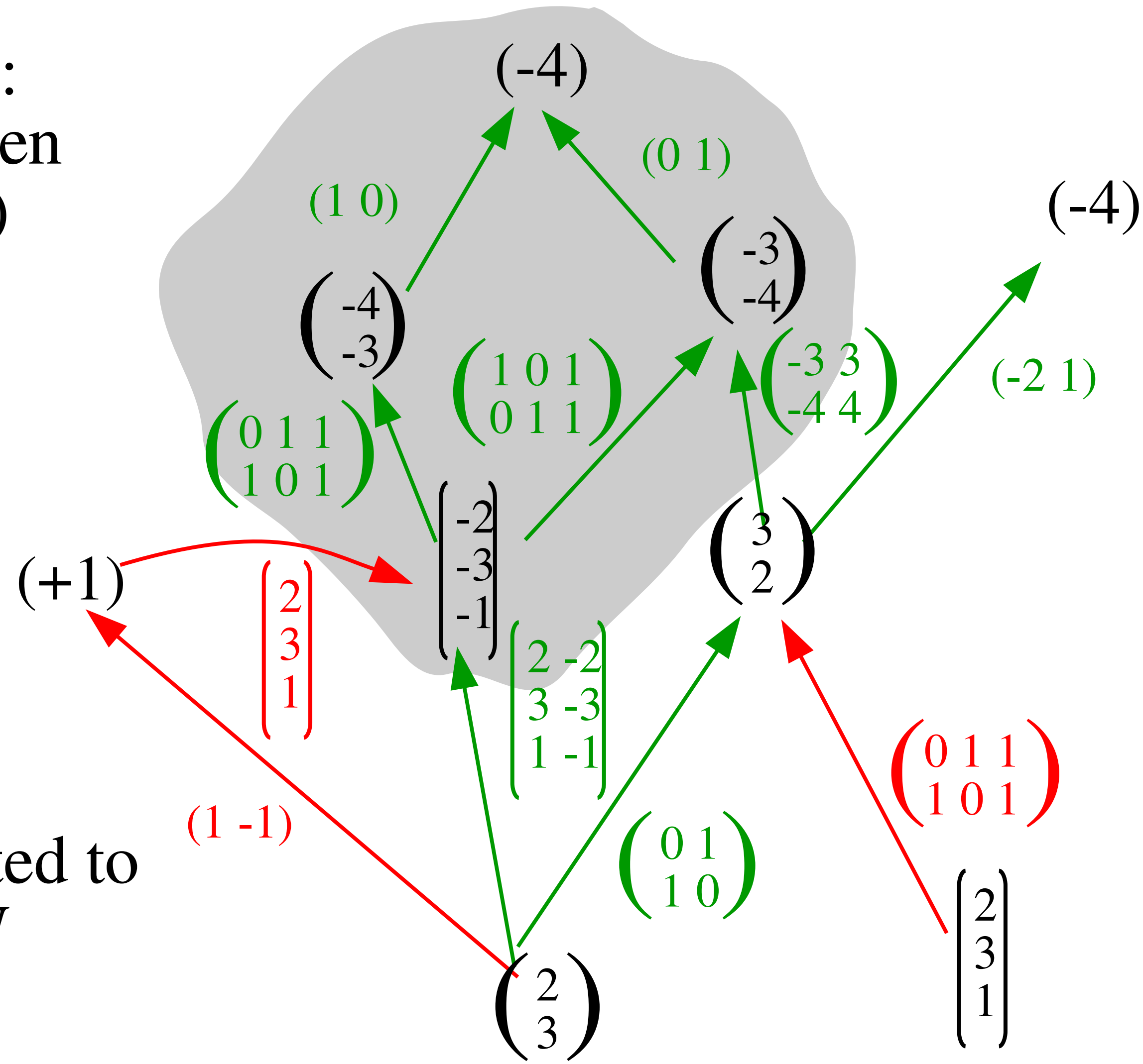



Thm: (R.) Consistency radius sets a lower bound on the distance to the nearest global section

*Data fusion* selects the nearest global section

# *Consistency radius* is monotonic

<u>Proposition</u>:
If $U \subseteq V$ then
$c(U) \leq c(V)$



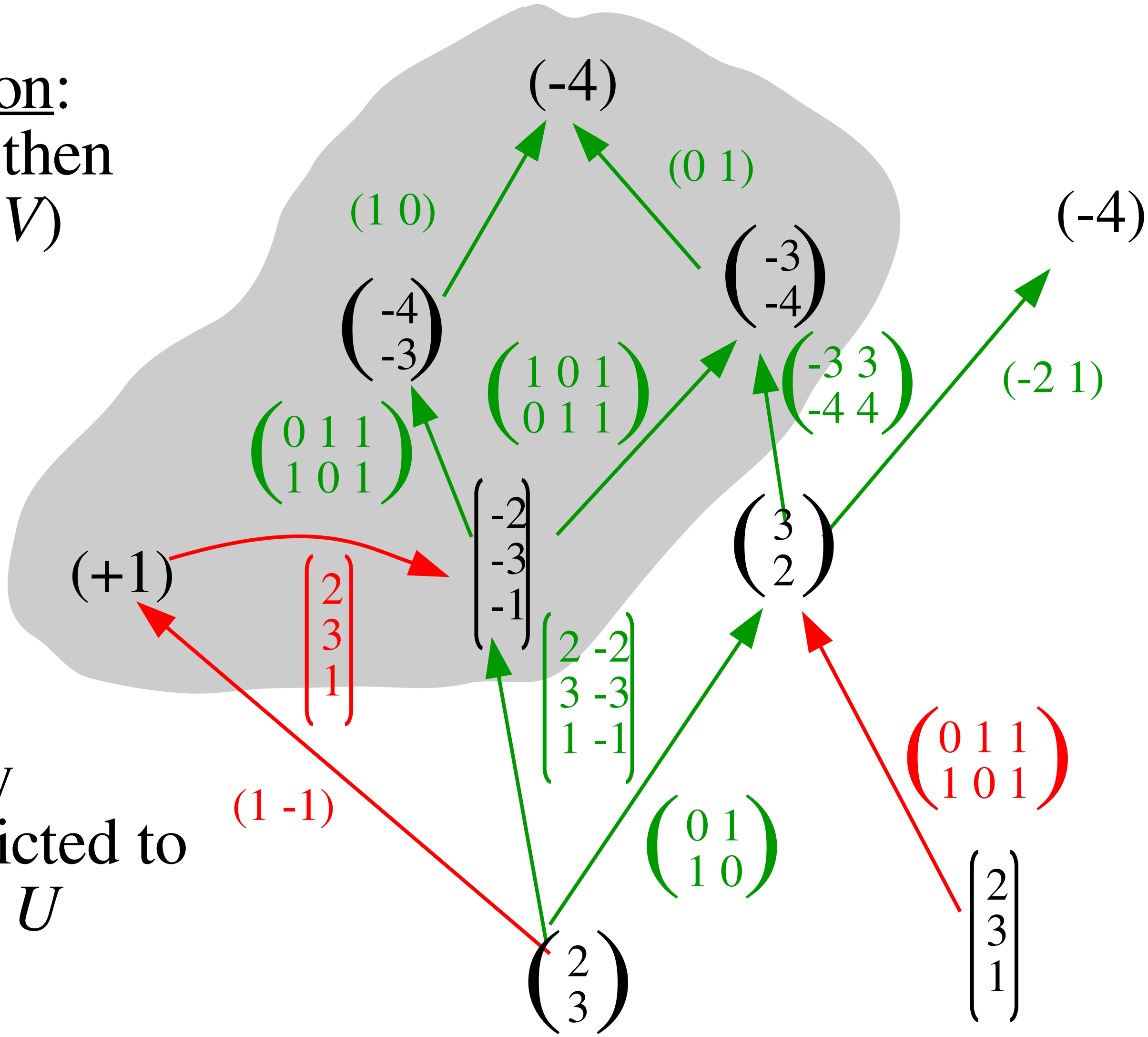

Consistency radius restricted to an open set $U$

Consistency Radius = 0

Note: lots more restrictions to check!

# *Consistency radius* is monotonic

Proposition:
If $U \subseteq V$ then
$c(U) \leq c(V)$

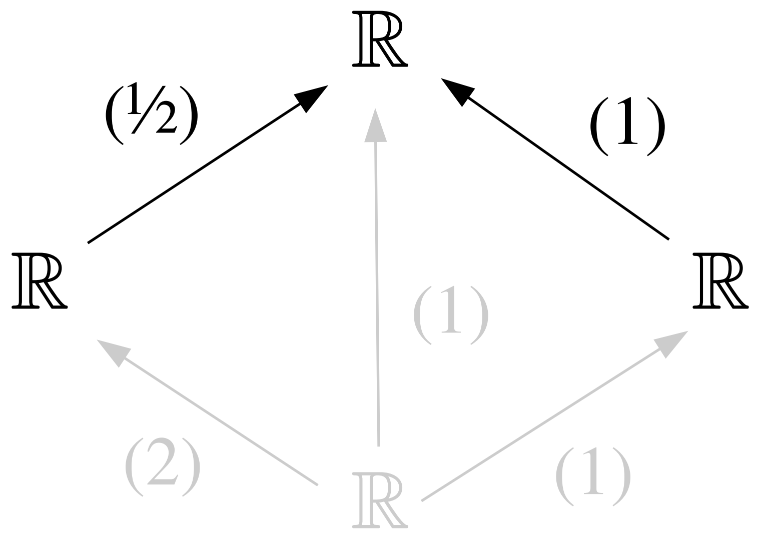



Consistency radius restricted to an open set $U$

Consistency Radius = $2\sqrt{14}$

Note: lots more restrictions to check!

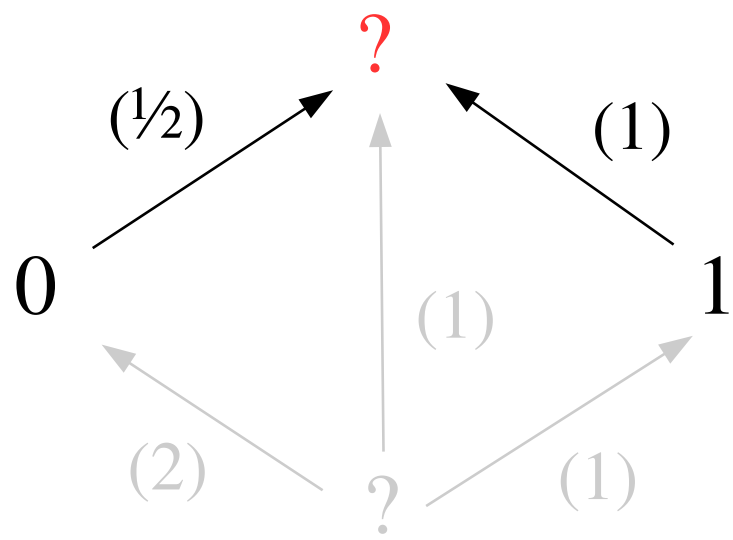
# *Consistency radius* optimization




NB: restrictions act by multiplication

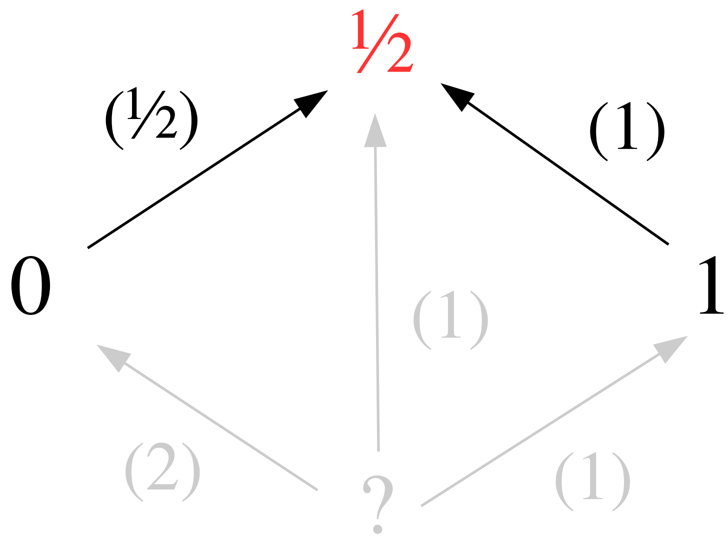
This is a sheaf on a small poset

# *Consistency radius* optimization



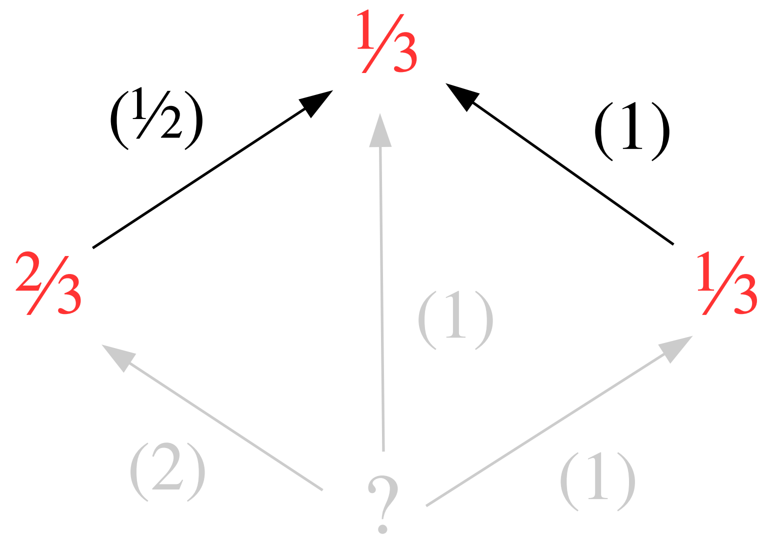

Here is an assignment supported on part of it

# *Consistency radius* optimization



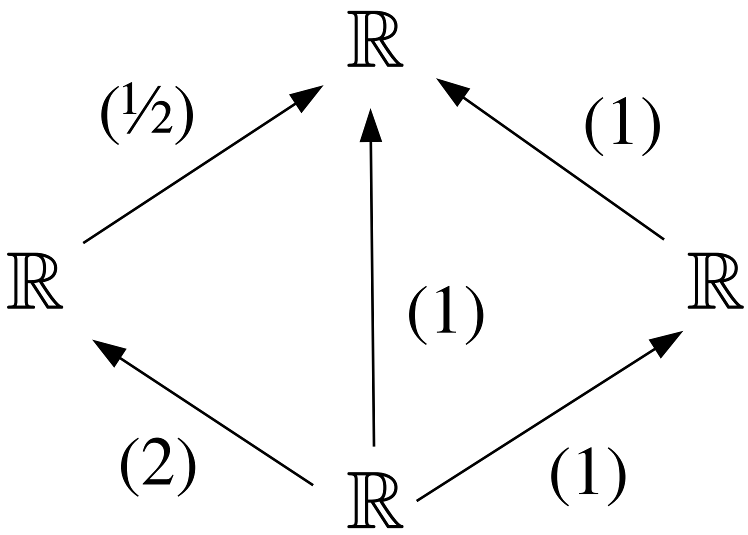

Minimizing the consistency radius when extending globally

# *Consistency radius* optimization



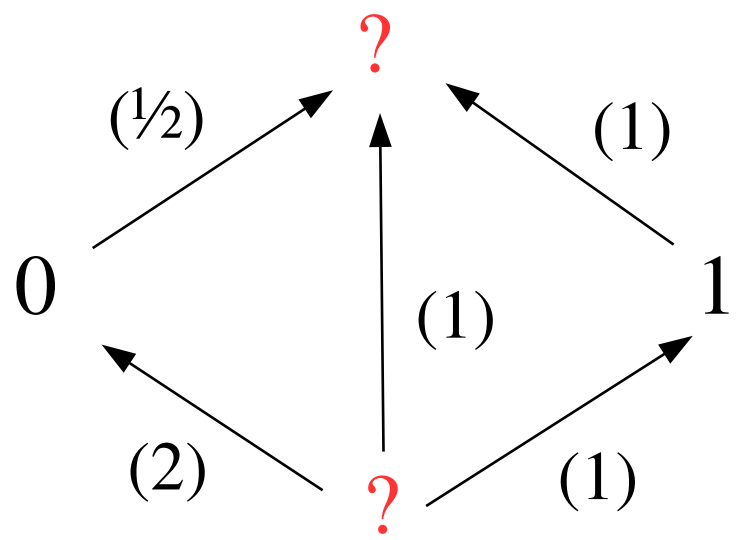

Here is the closest global section (everything can be changed)

# Extending to a larger sheaf...

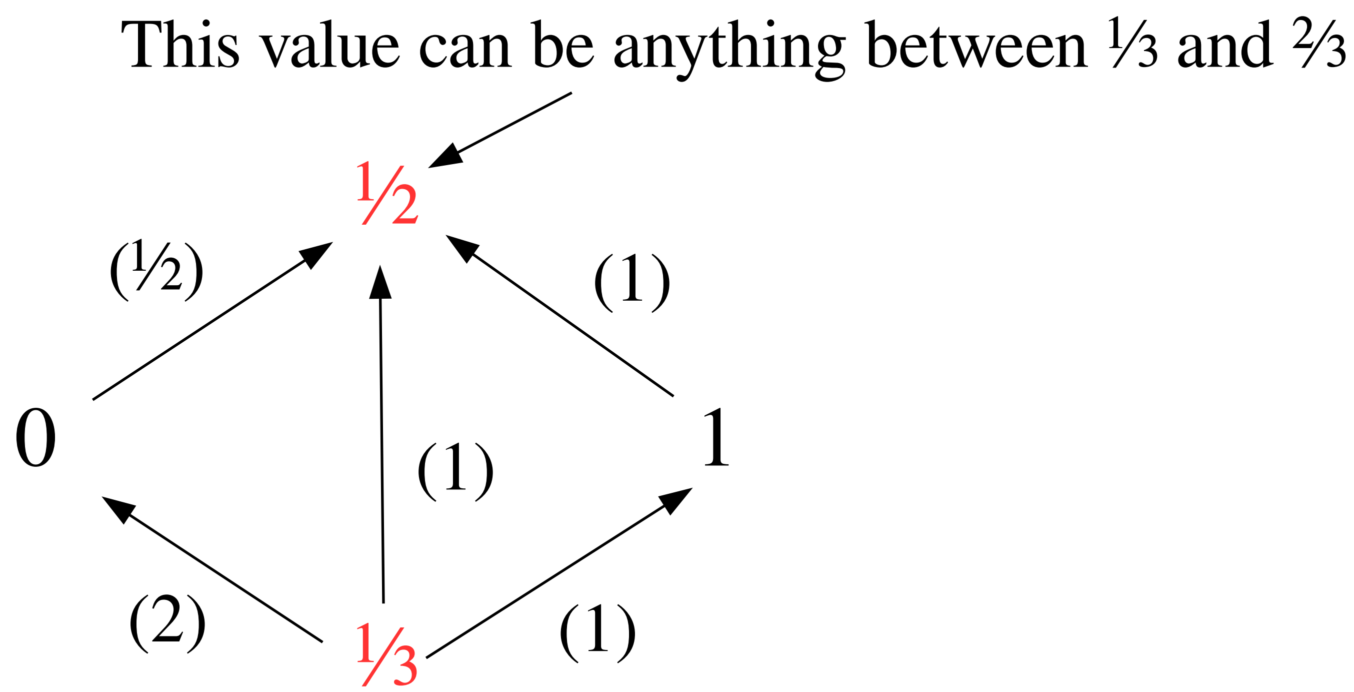



This is the full sheaf diagram including all open sets in the Alexandrov topology, not just the base

# Extending to a larger sheaf...



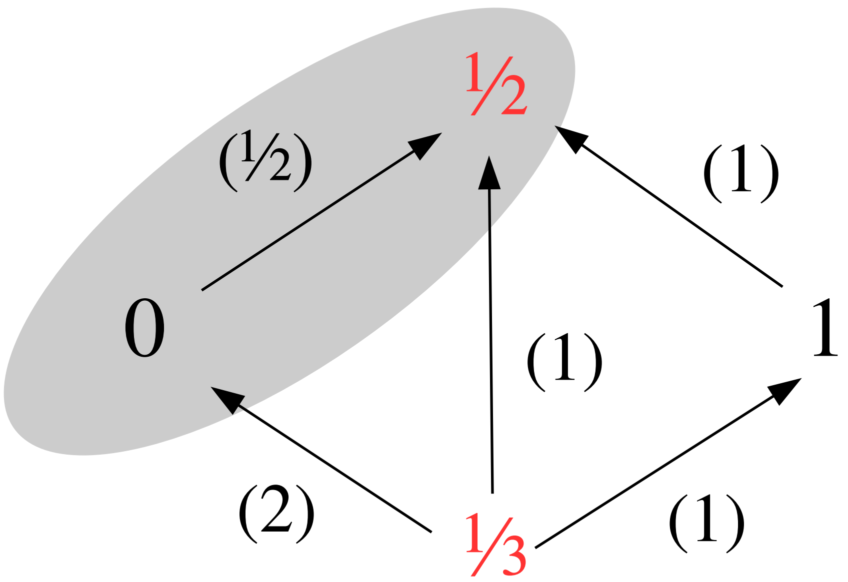
# Extending to a larger sheaf…




Minimizing the consistency radius when extending
The value on the intersection is no longer unique!

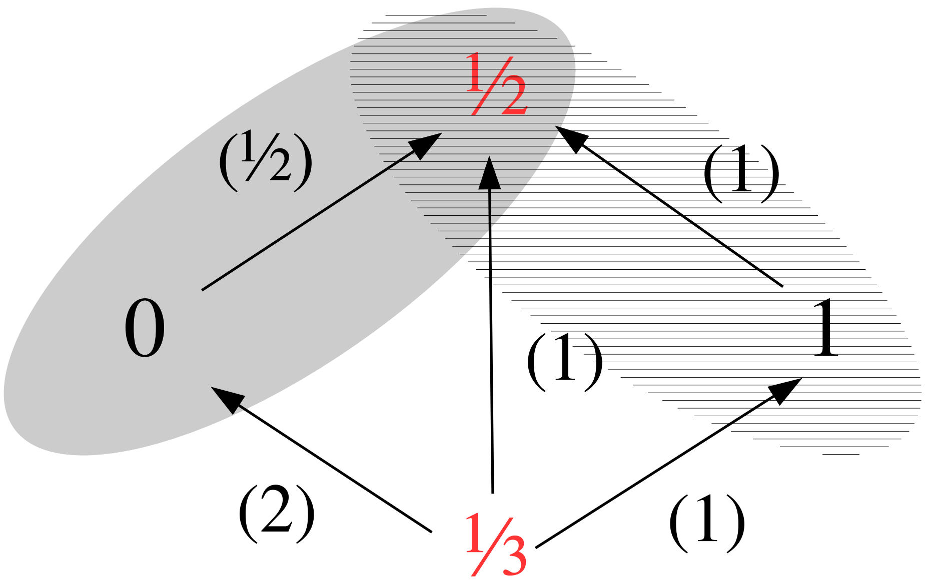
# Extending to a larger sheaf...

c($U$) = ½



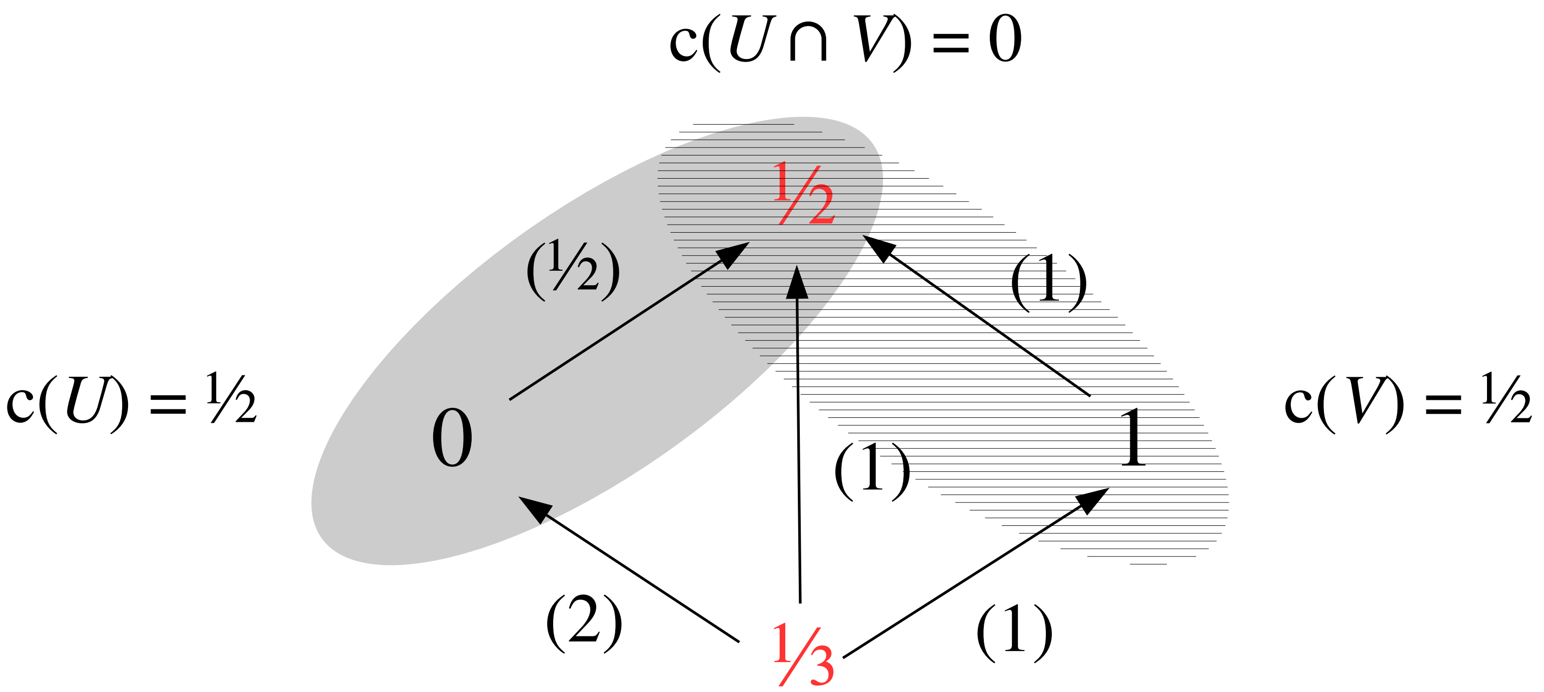
# Extending to a larger sheaf...

c($U$) = ½

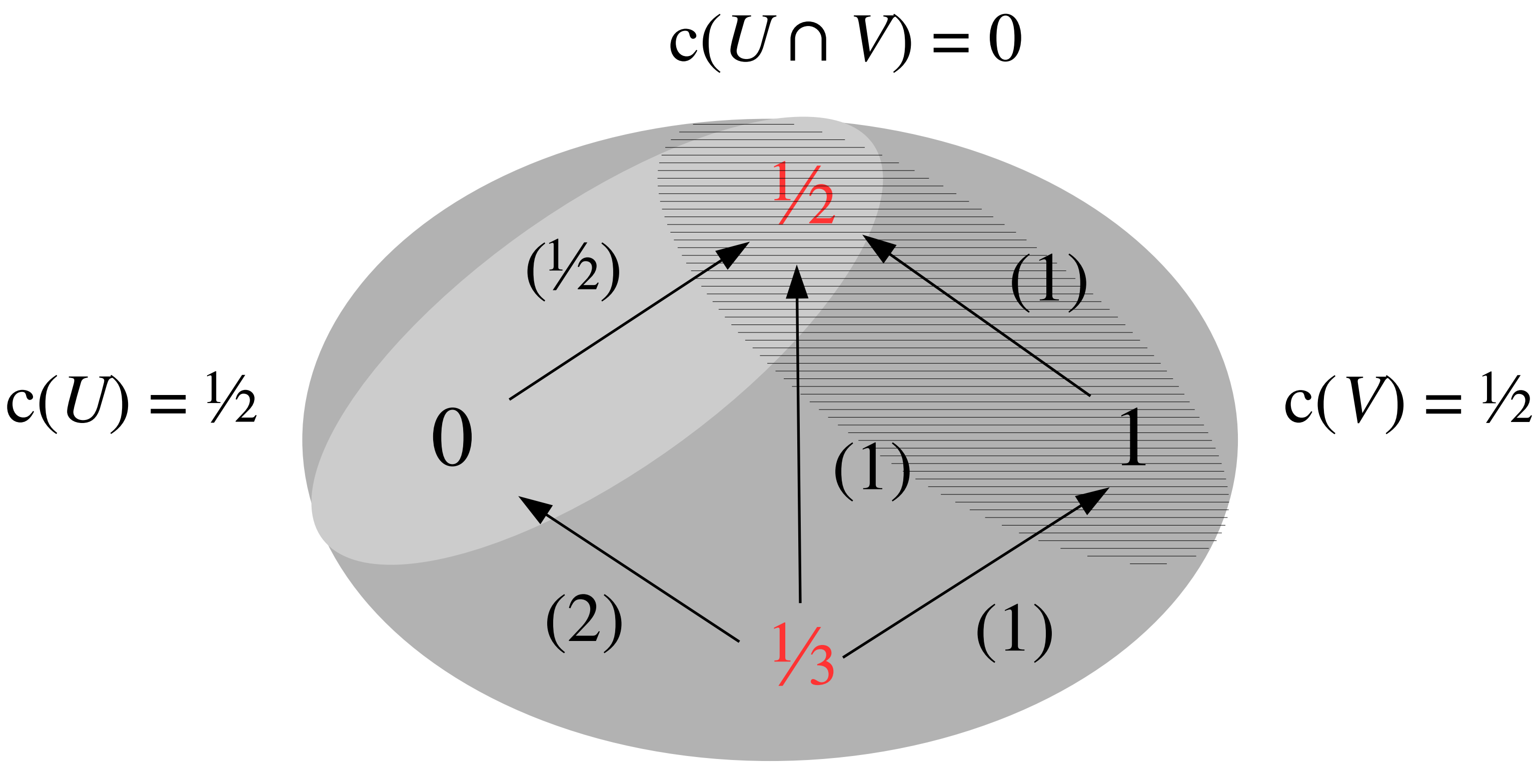



c($V$) = ½

# Extending to a larger sheaf...



# *Consistency radius* is not a measure




c(*U* ∪ *V*) = ⅔ < c(*U*) + c(*V*) – c(*U* ∩ *V*) = 1

Proposition:
"Local consistency of a Global assignment" **is a (loose) upper bound for** "Global consistency of a Local assignment"

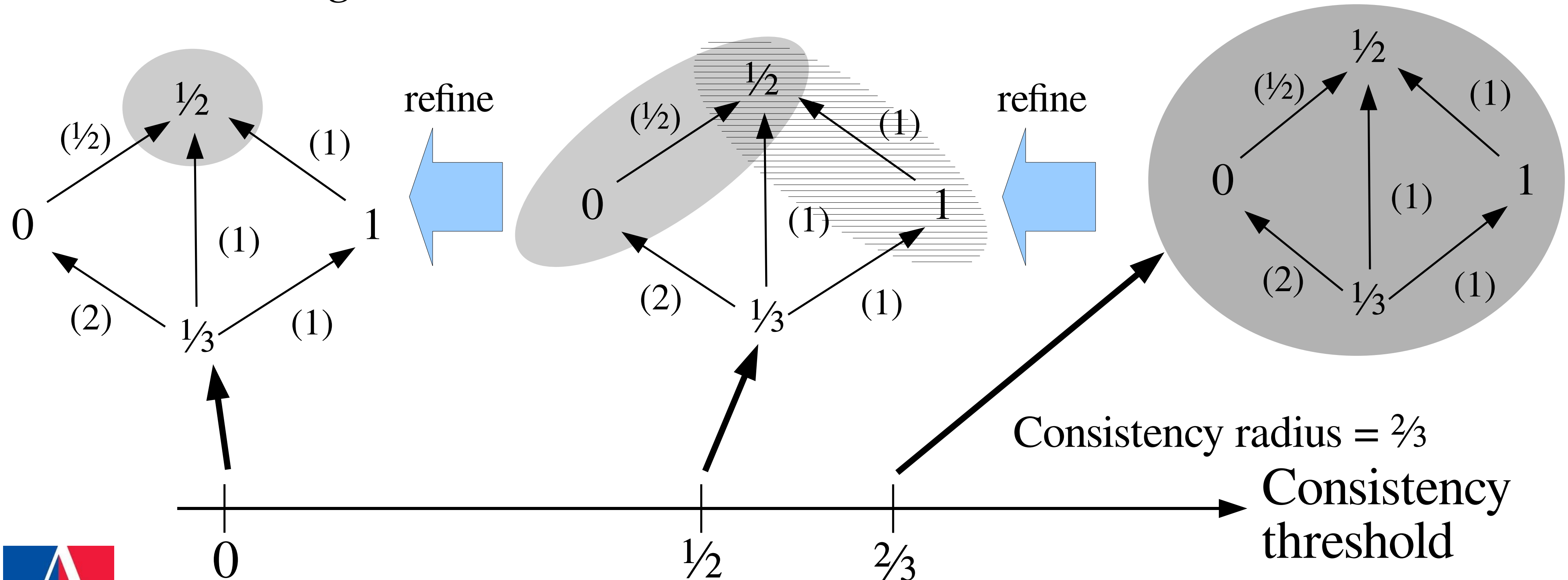
# Software!

- Computing local consistency radius

https://colab.research.google.com/drive/1hscWfilQFls_fOBSo3YtpVMQ-Tz2UNWn

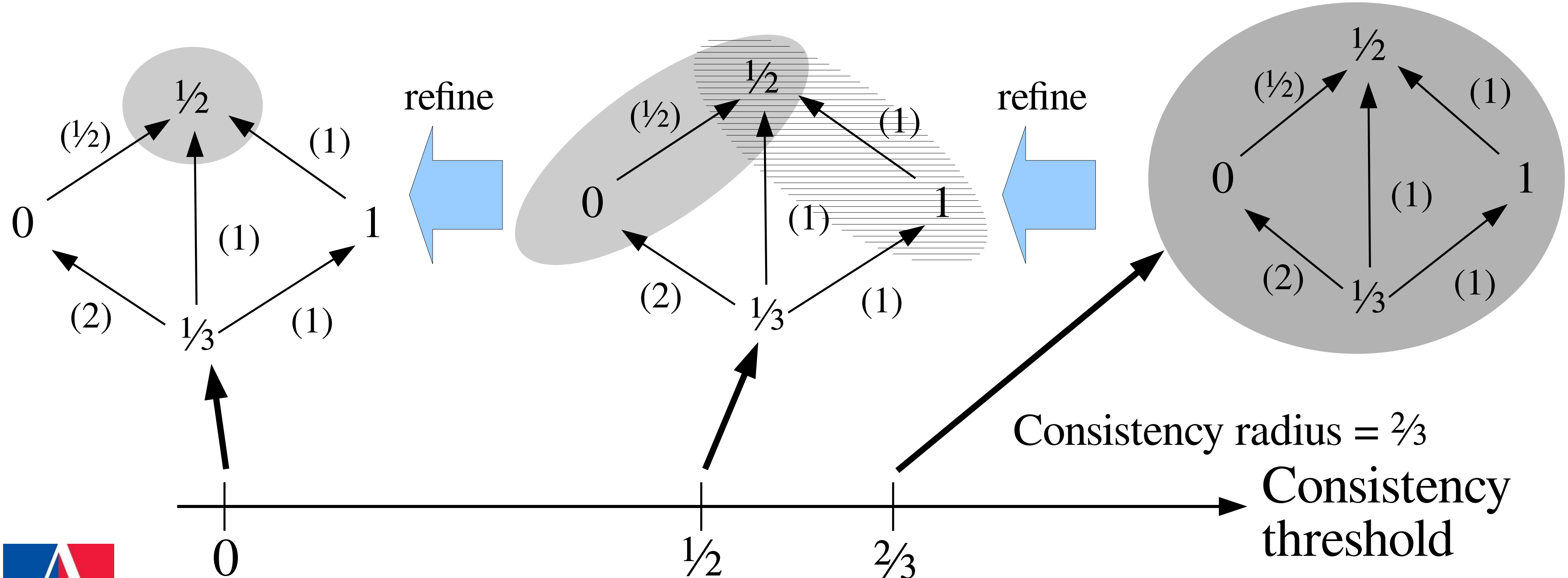
# The *consistency filtration*

- … assigns the set of open sets (open cover) with consistency less than a given threshold
- <u>Lemma</u>: consistency filtration **is itself a sheaf** of collections of open sets on (ℝ,≤). Restrictions in this sheaf are *cover coarsenings*.



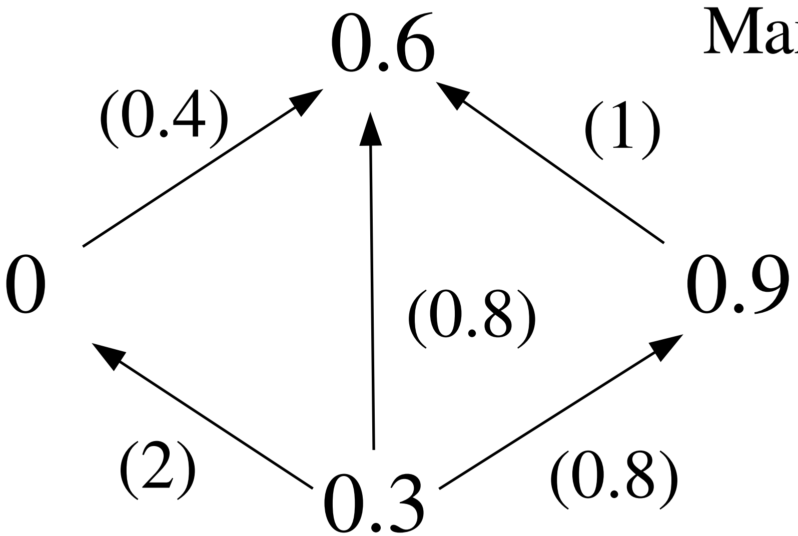
# Consistency filtration is natural

- Theorem: Consistency filtration is continuous under the an *interleaving distance*
- Theorem: Consistency filtration is also functorial
- (Note: the proof is quite intricate...)

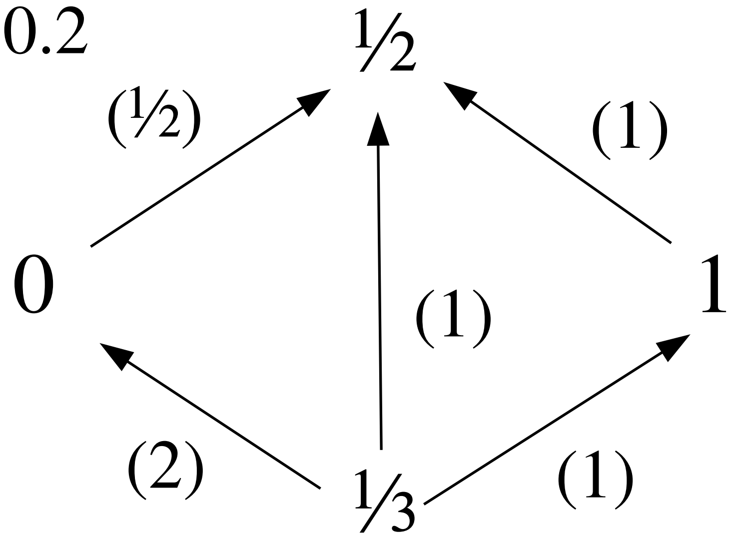


# A small perturbation …

- Perturbations allowed in both assignment **and** sheaf (subject to it staying a sheaf!)

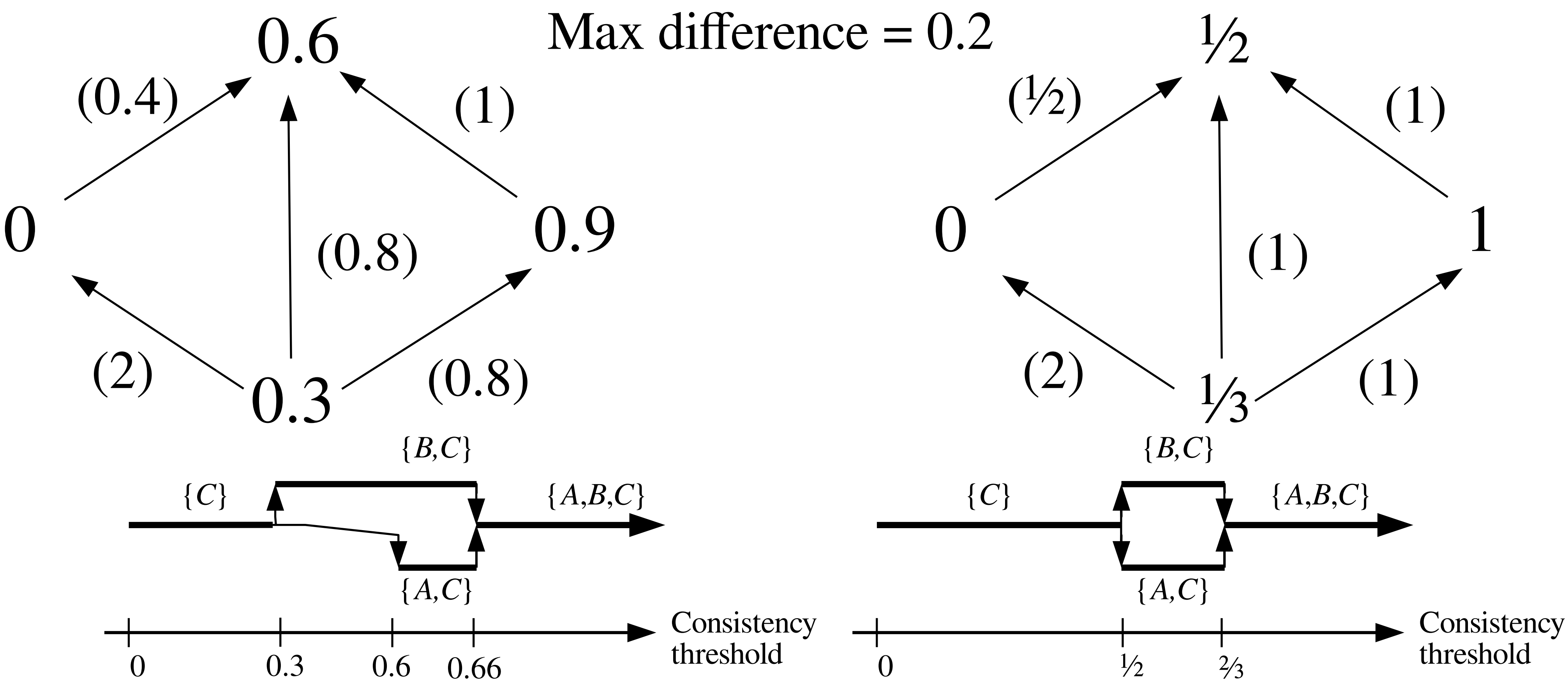



Max difference = 0.2



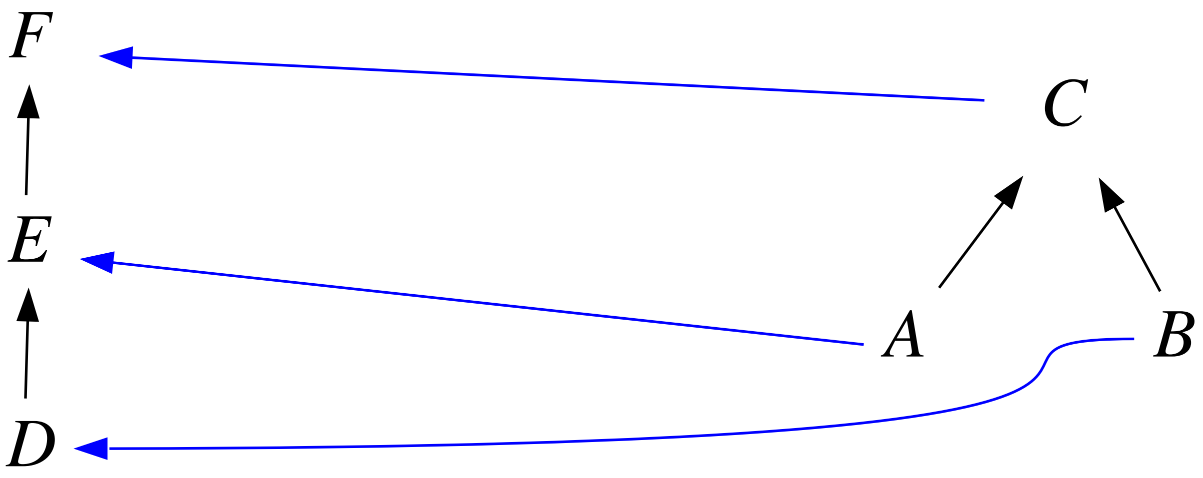
# A small perturbation …

- Compute consistency filtrations… they're similar



# A sheaf assignment *morphism* is ...

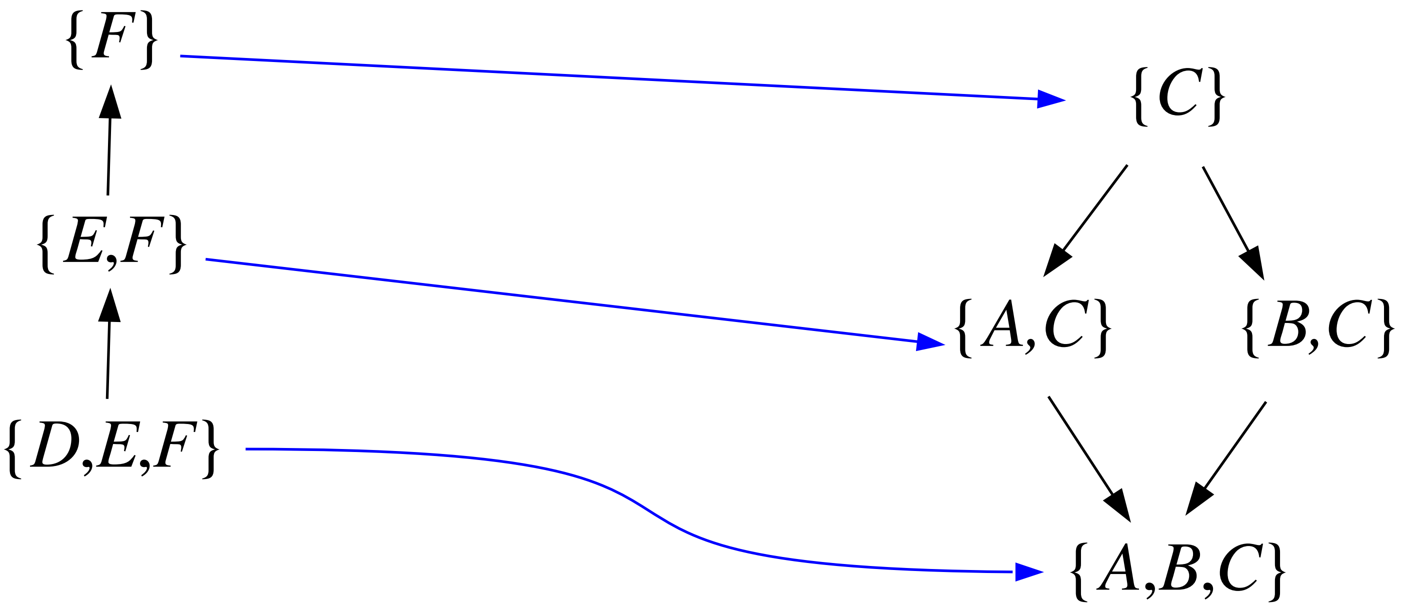
- … first, an order preserving map between base posets...



# A sheaf assignment *morphism* is ...

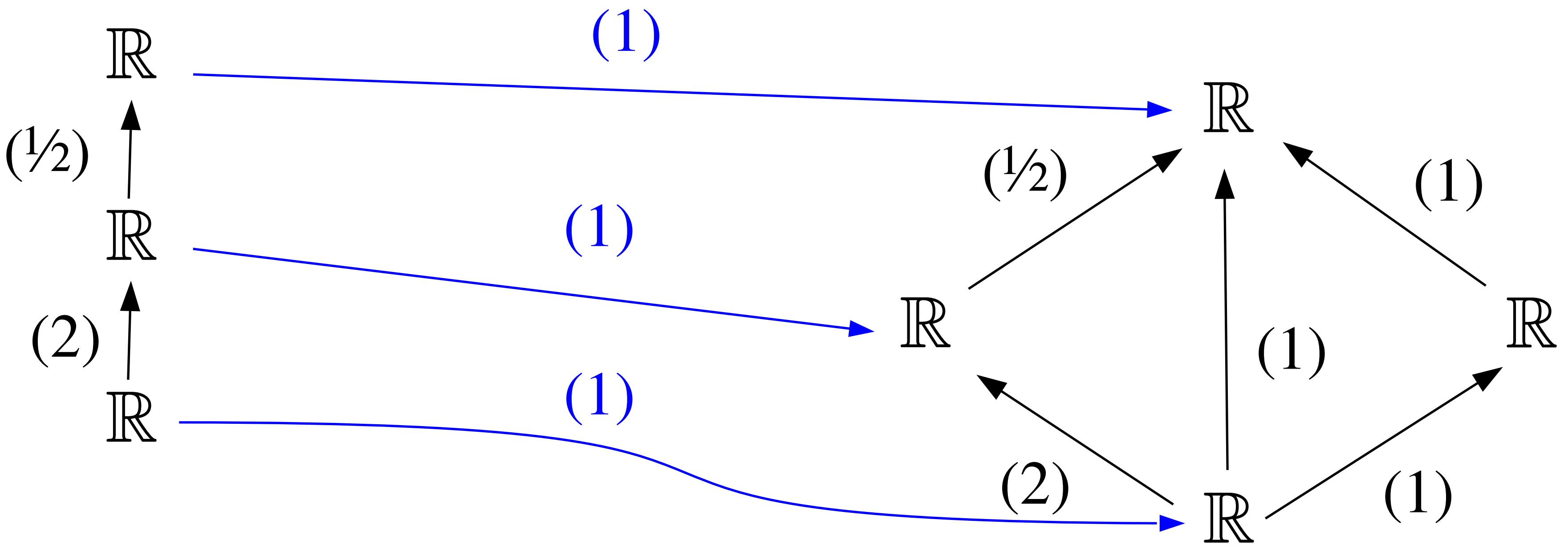
- … which is a continuous map …

  (preimages shown below)



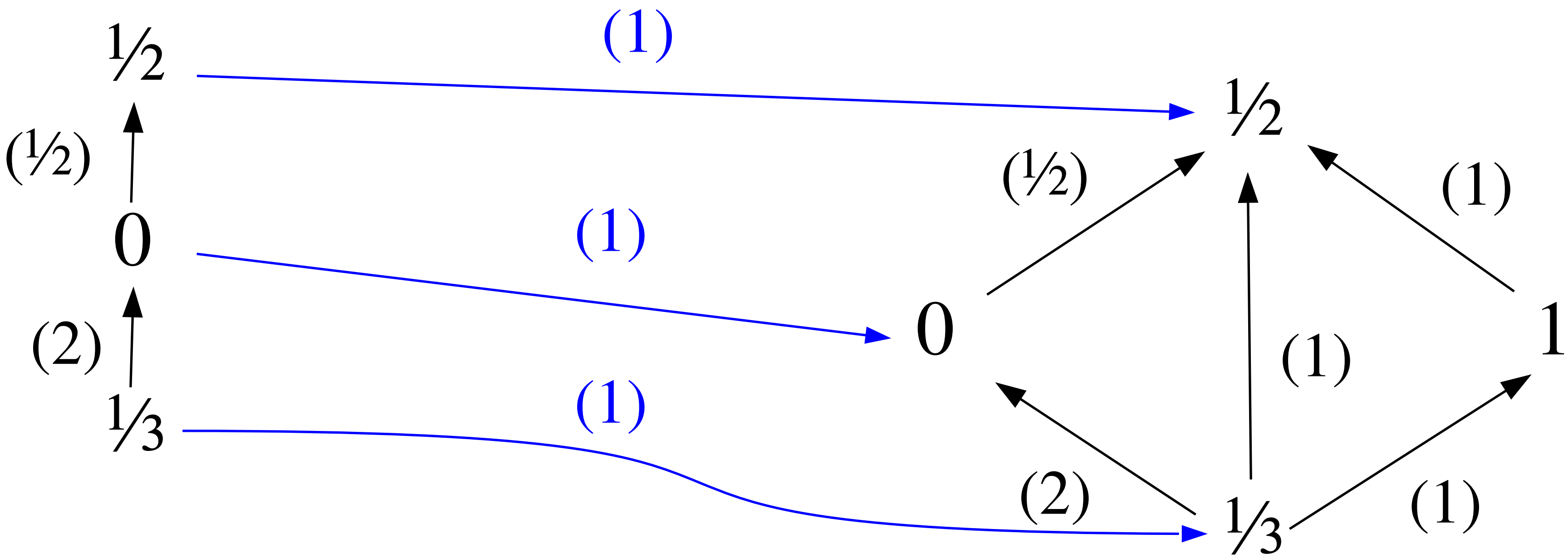
# A sheaf assignment *morphism* is ...

- … add to this, a commuting set of *component maps* for the two sheaves …



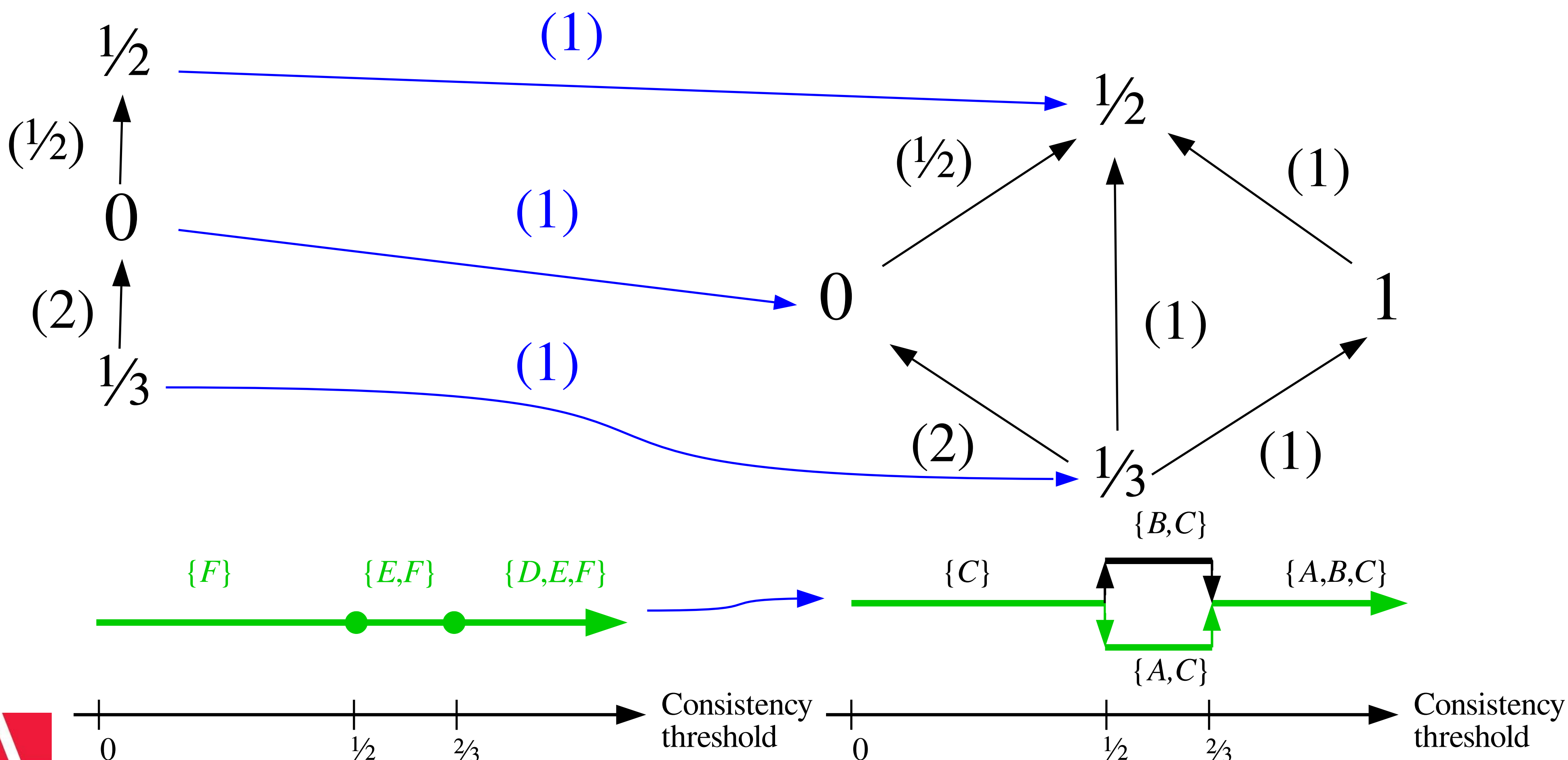
# A sheaf assignment *morphism* is ...

- … such that the assignments on both ends are preserved.



# Functoriality!

- Compute consistency filtrations, and all that's really needed is to align the open sets in the covers!



# Interpretation

- Sheaf: a **data structure** for modeling consistency
- Assignment: an **instance** of the data housed in a sheaf
- Consistency radius: **how well** do data and model agree?
- Consistency radius optimization: **predict** some missing or cleaner data
- Consistency filtration: **where** do data and model agree?