# An Optimal Architecture for Decentralized Control over Posets

Parikshit Shah    Pablo A. Parrilo

Laboratory for Information and Decision Systems
Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Stanford ISL - March 2011

# Motivation

- Many decision-making problems are large-scale and complex.
- Complexity, cost, physical constraints $\Rightarrow$ Decentralization.
- Fully distributed control is notoriously hard.
- A common underlying theme: flow of information.
- What are the right language and tools to think about flow of information?

**Contributions**

A framework to reason about information flow in terms of partially ordered sets (posets).

An architecture for decentralized control, based on Möbius inversion, with provable optimality properties.

# Motivation

- Many decision-making problems are large-scale and complex.
- Complexity, cost, physical constraints $\Rightarrow$ Decentralization.
- Fully distributed control is notoriously hard.
- A common underlying theme: flow of information.
- What are the right language and tools to think about flow of information?

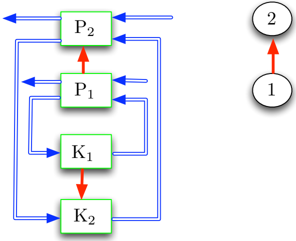
## Contributions

A framework to reason about information flow in terms of partially ordered sets (posets).

An architecture for decentralized control, based on Möbius inversion, with provable optimality properties.

# Motivation

- Many interesting examples can be unified in this framework.
- Example: Nested Systems [Voulgaris00].




- Emphasis: *Flow of information*. Can abstract away this flow of information to picture on right.
- Natural for problems of causal or hierarchical nature.

# Outline

- Basic Machinery: Posets and Incidence Algebras.
- Decentralized control problems and posets.
- $\mathcal{H}_2$ case: state-space solution
- Zeta function, Möbius inversion
- Controller architecture

# Partially ordered sets (posets)

### Definition

A poset $\mathcal{P} = (P, \preceq)$ is a set $P$ along with a binary relation $\preceq$ which satisfies for all $a, b, c \in P$:

1. $a \preceq a$ (reflexivity)
2. $a \preceq b$ and $b \preceq a$ implies $a = b$ (antisymmetry)
3. $a \preceq b$ and $b \preceq c$ implies $a \preceq c$ (transitivity).

- Will deal initially with finite posets (i.e. $|P|$ is finite).
- Will relate posets to decentralized control.

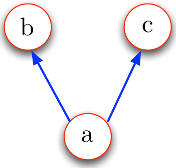
# Incidence Algebras

### Definition

The set of functions $f : P \times P \to \mathbb{Q}$ with the property that $f(x, y) = 0$ whenever $y \not\preceq x$ is called the incidence algebra $\mathcal{I}$.

- Concept developed and studied in [Rota64] as a unifying concept in combinatorics.
- For finite posets, elements of the incidence algebra can be thought of as matrices with a particular sparsity pattern.

# Example




$$\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ * & 0 & * \end{bmatrix} \end{array}$$

# Example

- Closure under addition and scalar multiplication.
- What happens when you multiply two such matrices?

$$\begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ * & 0 & * \end{bmatrix} \begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ * & 0 & * \end{bmatrix} = \begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ * & 0 & * \end{bmatrix}$$

- *Not a coincidence!*

# Example

- Closure under addition and scalar multiplication.
- What happens when you multiply two such matrices?

$$\begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ * & 0 & * \end{bmatrix} \begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ * & 0 & * \end{bmatrix} = \begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ * & 0 & * \end{bmatrix}$$

- *Not a coincidence!*

# Incidence Algebras

- Closure properties are true in general for all posets.

**Lemma**

*Let $\mathcal{P}$ be a poset and $\mathcal{I}$ be its incidence algebra. Let $A, B \in \mathcal{I}$ then:*

1. $c \cdot A \in \mathcal{I}$
2. $A + B \in \mathcal{I}$
3. $AB \in \mathcal{I}$.

*Thus the incidence algebra is an associative algebra.*

- A simple corollary: Since $I$ is in every incidence algebra, if $A \in \mathcal{I}$ and invertible, $A^{-1} \in \mathcal{I}$.
- Properties useful in Youla domain.

# Incidence Algebras

- Closure properties are true in general for all posets.

**Lemma**

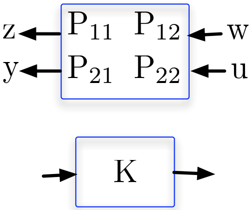
*Let $\mathcal{P}$ be a poset and $\mathcal{I}$ be its incidence algebra. Let $A, B \in \mathcal{I}$ then:*

1. $c \cdot A \in \mathcal{I}$
2. $A + B \in \mathcal{I}$
3. $AB \in \mathcal{I}$.

*Thus the incidence algebra is an associative algebra.*

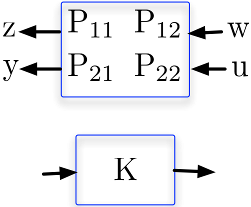
- A simple corollary: Since $I$ is in every incidence algebra, if $A \in \mathcal{I}$ and invertible, $A^{-1} \in \mathcal{I}$.
- Properties useful in Youla domain.

# Control problem



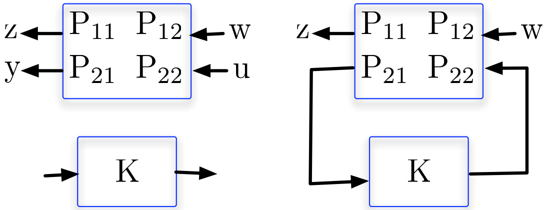

- A given matrix *P*.
- Design *K*.
- Interconnect *P* and *K*

# Control problem




- A given matrix $P$.
- Design $K$.
- Interconnect $P$ and $K$

# Control problem

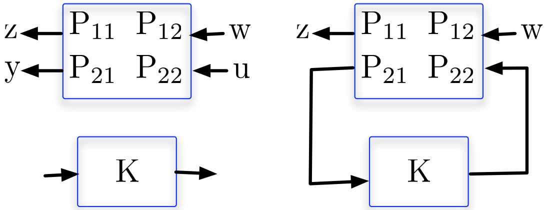



- A given matrix $P$.
- Design $K$.
- Interconnect $P$ and $K$

$$f(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}.$$

- Find "best" $K$.

# Control problem

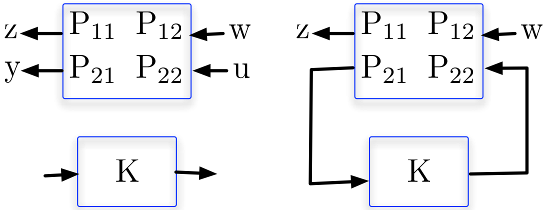



- A given matrix $P$.
- Design $K$.
- Interconnect $P$ and $K$

$$f(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}.$$

- Find “best” $K$.

# Control problem



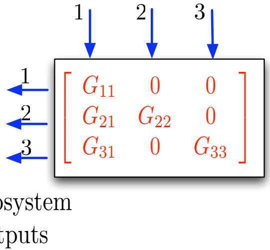

- A given matrix $P$.
- Design $K$.
- Interconnect $P$ and $K$

$$f(P,K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}.$$

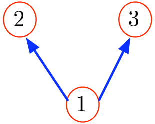
- Find "best" $K$.

# Modeling decentralized control problems using posets

- All the action happens at $P_{22} = G$. Focus here.
- $G$ (called the plant) interacts with the controller.
- Plant divided into subsystems:

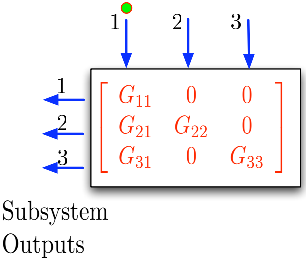


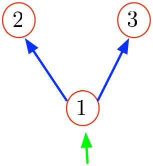

$$\begin{bmatrix} G_{11} & 0 & 0 \\ G_{21} & G_{22} & 0 \\ G_{31} & 0 & G_{33} \end{bmatrix}$$



# Modeling decentralized control problems using posets

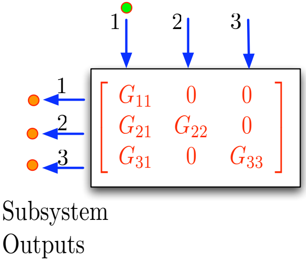
- Let $G$ be the transfer function matrix of the plant. We divide up the plant into subsystems:

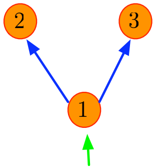


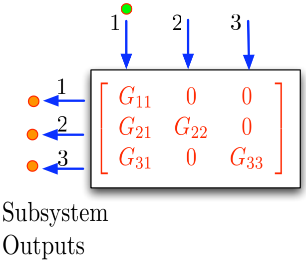
# Modeling decentralized control problems using posets

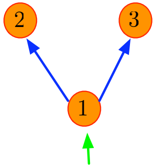
- Let $G$ be the transfer function matrix of the plant. We divide up the plant into subsystems:



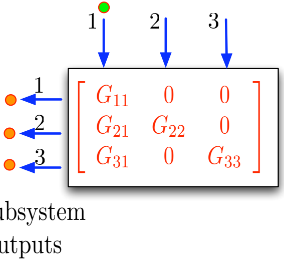
# Modeling decentralized control problems using posets

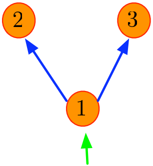



- Denote this by $1 \preceq 2$ and $1 \preceq 3$.
- Subsystems 2 and 3 are in cone of influence of 1
- This relationship is a causality relation between subsystems.
- We call systems with $G \in \mathcal{I}$ poset-causal systems.

# Modeling decentralized control problems using posets




- Denote this by $1 \preceq 2$ and $1 \preceq 3$.
- Subsystems 2 and 3 are in cone of influence of 1
- This relationship is a causality relation between subsystems.
- We call systems with $G \in \mathcal{I}$ poset-causal systems.

# Controller Structure

- Given a poset causal plant $G \in \mathcal{I}$.
- Decentralization constraint: mirror the information structure of the plant.
- In other words we want poset-causal $K \in \mathcal{I}$.
- Similar causality interpretation.
- Intuitively, $i \preceq j$ means subsystem $j$ is more information rich.
- The poset arranges the subsystems according to the amount of information richness.

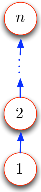
# Controller Structure

- Given a poset causal plant $G \in \mathcal{I}$.
- Decentralization constraint: mirror the information structure of the plant.
- In other words we want poset-causal $K \in \mathcal{I}$.
- Similar causality interpretation.
- Intuitively, $i \preceq j$ means subsystem $j$ is more information rich.
- The poset arranges the subsystems according to the amount of information richness.

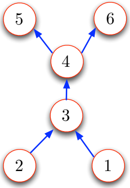
# Examples of poset systems

- Independent subsystems


- Nested systems




- Closures of directed acyclic graphs



# General framework

Goal is to capture what is essential about causal decision-making.

- Elements of the poset do not necessarily have to represent only subsystems.
  - “Standard” case discussed earlier corresponds to the product of the spatial interaction poset (space) and a linear chain (time)
  - Posets model branching time, nondeterminism, etc.
  - Posets in space-time (e.g., distributed systems)
- Controller structure does not necessarily have to mirror plant.
  - Generalizations via *Galois connections*

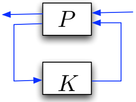
# Literature review

- Classical work: Witsenhausen, Radner, Ho-Chu.
- Mullans-Elliot (1973), linear systems on partially ordered time sets
- Voulgaris (2000), showed that a wide class of distributed control problems became convex through a Youla-type reparametrization.
- Rotkowitz-Lall (2002) introduced *quadratic invariance* (QI) an important unifying concept for convexity in decentralized control.
- Poset framework introduced in Shah-P. (2008). Special case of QI, with richer and better understood algebraic structure.
- Swigart-Lall (2010) gave a state-space solution for the two-controller case, via a spectral factorization approach.
- Shah-P. (2010), provided a full solution for all posets, with controller degree bounds. Separability a key idea, which is missing in past work. Introduced simple Möbius-based architecture (in slightly different form).

# Optimal Control Problem

Given a system $P$ with plant $G$, find a stabilizing controller $K \in \mathcal{I}$.

$$\begin{array}{ll} \text{minimize}_K & ||f(P,K)|| \\ \text{subject to} & K \text{ stabilizes } P \\ & K \in \mathcal{I}. \end{array}$$




- Here $f(P,K) = P_{11} + P_{12}K(I - GK)^{-1}P_{21}$ is the closed loop transfer function.
- Problem is nonconvex.
- Standard approach: reparametrize the problem by getting rid of the nonconvex part of the objective.

# Convex reparametrization

- "Youla domain" technique: define $R = K(I - GK)^{-1}$.

$$\begin{array}{ll} \text{minimize} & \|\hat{P}_{11} + \hat{P}_{12} R \hat{P}_{21}\| \\ \text{subject to} & R \text{ stable} \\ & R \in \mathcal{I}. \end{array}$$

  Algebraic structure of $\mathcal{I}$ allows to rewrite a convex constraint in $K$ into a convex constraint in $R$.

- Main difficulty: Infinite dimensional problem.
- Can be approximated by various techniques, but there are drawbacks.
- Desire state-space techniques. Advantages:
  1. Computationally efficient
  2. Degree bounds
  3. Provide insight into structure of optimal controller.

# State-Space Setup

- Have state feedback system:

$$\begin{aligned} x[t+1] &= Ax[t] + Bu[t] + w[t] \\ y[t] &= x[t] \\ z[t] &= Cx[t] + Du[t] \end{aligned}$$

- Wish to find controller $u = Kx$ which is stabilizing and optimal.

$$\begin{aligned} &\text{Min}_K \|P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}\|^2 \\ &K \in \mathcal{I} \\ &K \text{ stabilizing.} \end{aligned}$$

- Key property we exploit: separability of the $\mathcal{H}_2$ norm.

# $\mathcal{H}_2$ Optimal Control

- Recall Frobenius norm:

$$\|H\|_F^2 = \text{Trace}(H^T H).$$

- $\mathcal{H}_2$ norm is its extension to operators.
- Solution to optimal centralized problem standard.
- Based on algebraic Riccati equations:

$$X = C^T C + A^T XA - A^T XB(D^T D + B^T XB)^{-1} B^T XA$$
$$K = (D^T D + B^T XB)^{-1} B^T XA.$$

# Decentralized Control Problem

- System poset causal: $A, B \in \mathcal{I}(\mathcal{P})$.
- Solve:

$$\begin{aligned} \text{minimize}_K & \|P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}\|^2 \\ & K \in \mathcal{I} \\ & K \text{ stabilizing.} \end{aligned}$$

- Due to state-feedback: $P_{21} = (zI - A)^{-1}$.
- Define $Q := K(I - GK)^{-1}P_{21}$.
- Problem reduces to:

$$\begin{aligned} \text{minimize}_Q & \|P_{11} + P_{12}Q\|^2 \\ & Q \in \mathcal{I} \\ & Q \text{ stabilizing.} \end{aligned}$$

# $\mathcal{H}_2$ Decomposition Property

- Let $G = [G_1, \dots G_k]$.

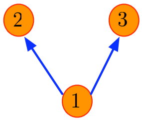
$$\|G\|^2 = \sum_{i=1}^{k} \|G_i\|^2.$$

- This separability property is the key feature we exploit.

## Example

$$\begin{array}{ll} \min & \left\| P_{11} + P_{12} \begin{bmatrix} Q_{11} & 0 & 0 \\ Q_{21} & Q_{22} & 0 \\ Q_{31} & 0 & Q_{33} \end{bmatrix} \right\|^2 \\ \text{s.t.} & Q \text{ stabilizing.} \end{array}$$

$$\begin{array}{ll} \min. & \left\| P_{11}(1) + P_{12} \begin{bmatrix} Q_{11} \\ Q_{21} \\ Q_{31} \end{bmatrix} \right\|^2 + \|P_{11}(2) + P_{12}(2) Q_{22}\|^2 \\ & + \|P_{11}(3) + P_{12}(3) Q_{33}\|^2 \\ \text{s.t.} & Q \text{ stabilizing.} \end{array}$$



# $\mathcal{H}_2$ State Space Solution

This decomposition idea extends to all posets.

**Theorem (Shah-P., CDC2010)**

*Problem can be reduced to decoupled problems:*

$$\begin{array}{rl} \textit{minimize} & \|P_{11}(j) + P_{12}(\uparrow j)Q^{\uparrow j}\|^2 \\ \textit{subject to} & Q^{\uparrow j} \textit{ stabilizing} \\ & \textit{for all } j \in P. \end{array}$$

- Optimal $Q$ can be obtained by solving a set of decoupled centralized sub-problems.
- Each sub-problem requires solution of a Riccati equation.

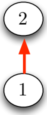
# $\mathcal{H}_2$ State Space Solution

- Can recover $K$ from optimal $Q$.
- $Q$ and $K$ are in bijection, $K = QP_{21}^{-1}(I + P_{22}QP_{21}^{-1})^{-1}$.
- Further analysis gives:
  1. Explicit state-space formulae.
  2. Controller degree bounds.
  3. Insight into structure of optimal controller.

# General Controller Architecture

Great. We solved the problem for all posets! But, what is the structure here?

- Swigart-Lall (2010) had a nice interpretation for the two-controller case, in terms of the first controller estimating the state of the second subsystem.




- No “obvious” generalizations:
  - In general, do not have enough information to predict upstream states. Also, there may be incomparable states.
  - More importantly, too many predictions from downstream! How to combine them?

# General Controller Architecture

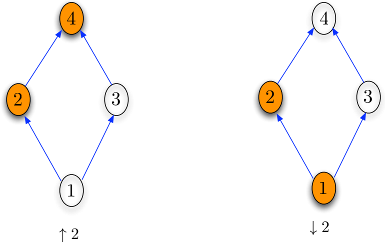
- What is the "right" architecture?
- Ingredients:
  1. Lower sets and upper sets
  2. Local variables (partial state predictions)
  3. Zeta function and Möbius function.
- Simple separation principle
- Optimality of architecture for $\mathcal{H}_2$.

# General Controller Architecture

- What is the “right” architecture?
- Ingredients:
  1. Lower sets and upper sets
  2. Local variables (partial state predictions)
  3. Zeta function and Möbius function.
- Simple separation principle
- Optimality of architecture for $\mathcal{H}_2$.

# Lower sets and upper sets

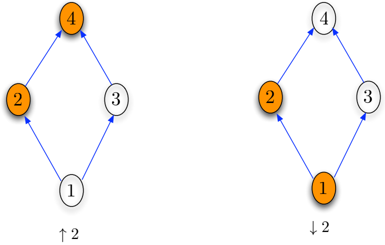
- Each "node" in $\mathcal{P}$ is a subsystem with state $x_i$ and input $u_i$.
- Lower set: $\downarrow p = \{q \mid q \preceq p\}$.
- Corresponds to "downstream" known information.




- Upper set: $\uparrow p = \{q \mid p \preceq q\}$.
- Corresponds to "upstream" unknown information.
- $u_i$ has access to $x_j$ for $j \in \downarrow i$ (downstream).

# Lower sets and upper sets

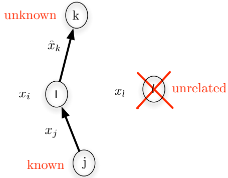
- Each "node" in $\mathcal{P}$ is a subsystem with state $x_i$ and input $u_i$.
- Lower set: $\downarrow p = \{q \mid q \preceq p\}$.
- Corresponds to "downstream" known information.



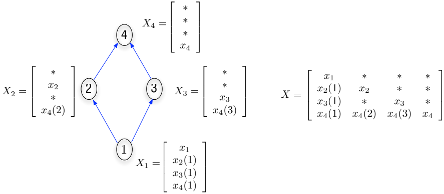

- Upper set: $\uparrow p = \{q \mid p \preceq q\}$.
- Corresponds to "upstream" unknown information.
- $u_i$ has access to $x_j$ for $j \in \downarrow i$ (downstream).

# Local Variables

- Overall state $x$ and input $u$ are global variables.
- Subsystems carry local copies.



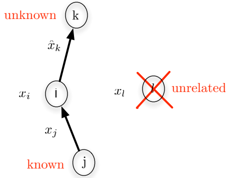

- Local variable $X_i : \uparrow i \to \mathbb{R}$.
- Can think of it as a vector in $\mathbb{R}^{|P|}$



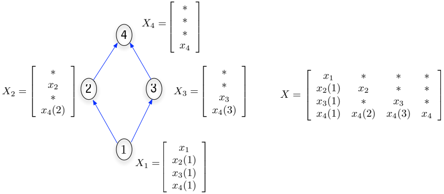

- Two local variables of interest:
  1. $X$: $X_{ij} = x_i(j)$ is the (partial) prediction of state $x_i$ at subsystem $j$.
  2. $U$: $U_{ij} = u_i(j)$ is the (partial) prediction of input $u_i$ at subsystem $j$.

# Local Variables

- Overall state $x$ and input $u$ are global variables.
- Subsystems carry local copies.



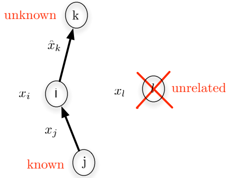

- Local variable $X_i : \uparrow i \to \mathbb{R}$.
- Can think of it as a vector in $\mathbb{R}^{|P|}$



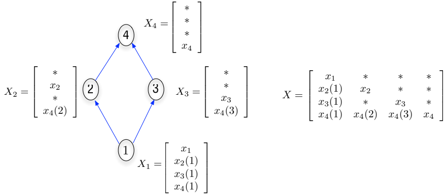

- Two local variables of interest:
  1. $X$: $X_{ij} = x_i(j)$ is the (partial) prediction of state $x_i$ at subsystem $j$.
  2. $U$: $U_{ij} = u_i(j)$ is the (partial) prediction of input $u_i$ at subsystem $j$.

# Local Variables

- Overall state $x$ and input $u$ are global variables.
- Subsystems carry local copies.



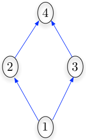

- Local variable $X_i : \uparrow i \to \mathbb{R}$.
- Can think of it as a vector in $\mathbb{R}^{|P|}$




- Two local variables of interest:
  1. $X$: $X_{ij} = x_i(j)$ is the (partial) prediction of state $x_i$ at subsystem $j$.
  2. $U$: $U_{ij} = u_i(j)$ is the (partial) prediction of input $u_i$ at subsystem $j$.

# Local Products

- Local gain: $G(i) : \uparrow i \times \uparrow i \to \mathbb{R}$. Think of it as zero-padded matrix:




$$G(2) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & G_{22} & 0 & G_{24} \\ 0 & 0 & 0 & 0 \\ 0 & G_{42} & 0 & G_{44} \end{bmatrix}$$

- Define $\mathbf{G} = \{G(1), \ldots, G(s)\}$.
- Local Product: $\mathbf{G} \circ X$ defined columnwise via:

$$(\mathbf{G} \circ X)_i = G(i)X_i.$$

- If $Y = \mathbf{G} \circ X$, then local variables $(X_i, Y_i)$ decoupled.

# Zeta and Möbius

For any poset $\mathcal{P}$, two distinguished elements of its incidence algebra:

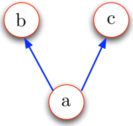
- The *Zeta* matrix is

$$\zeta_{\mathcal{P}}(x, y) = \begin{cases} 1, & \text{if } y \preceq x \\ 0, & \text{otherwise} \end{cases}$$

- Its inverse is the *Möbius* matrix of the poset:

$$\mu_{\mathcal{P}} = \zeta_{\mathcal{P}}^{-1}.$$

E.g., for the poset below, we have:




$$\zeta_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \qquad \mu_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

# Zeta and Möbius

For any poset $\mathcal{P}$, two distinguished elements of its incidence algebra:

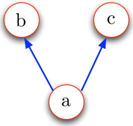
- The *Zeta* matrix is

$$\zeta_{\mathcal{P}}(x, y) = \begin{cases} 1, & \text{if } y \preceq x \\ 0, & \text{otherwise} \end{cases}$$

- Its inverse is the *Möbius* matrix of the poset:

$$\mu_{\mathcal{P}} = \zeta_{\mathcal{P}}^{-1}.$$

E.g., for the poset below, we have:




$$\zeta_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \qquad \mu_{\mathcal{P}} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

# Möbius inversion

Given $f : P \to \mathbb{Q}$, we can define

$$(\zeta f)(x) = \sum_y \zeta(x,y) f(y), \qquad (\mu f)(x) = \sum_y \mu(x,y) f(y).$$

These operations are obviously inverses of each other.

For our example:

$$\zeta(a_1, a_2, a_3) = (a_1, a_1 + a_2, a_1 + a_3), \qquad \mu(b_1, b_2, b_3) = (b_1, b_2 - b_1, b_3 - b_1).$$

**Möbius inversion formula**

$$g(y) = \sum_{x \preceq y} h(x) \qquad \Leftrightarrow \qquad h(y) = \sum_{x \preceq y} \mu(x,y) g(x)$$

# Möbius inversion

Given $f : P \to \mathbb{Q}$, we can define

$$(\zeta f)(x) = \sum_y \zeta(x,y) f(y), \qquad (\mu f)(x) = \sum_y \mu(x,y) f(y).$$

These operations are obviously inverses of each other.

For our example:

$$\zeta(a_1, a_2, a_3) = (a_1, a_1 + a_2, a_1 + a_3), \qquad \mu(b_1, b_2, b_3) = (b_1, b_2 - b_1, b_3 - b_1).$$

**Möbius inversion formula**

$$g(y) = \sum_{x \preceq y} h(x) \quad \Leftrightarrow \quad h(y) = \sum_{x \preceq y} \mu(x,y) g(x)$$

# Möbius inversion

Given $f : P \to \mathbb{Q}$, we can define

$$(\zeta f)(x) = \sum_y \zeta(x,y) f(y), \qquad (\mu f)(x) = \sum_y \mu(x,y) f(y).$$

These operations are obviously inverses of each other.

For our example:

$$\zeta(a_1, a_2, a_3) = (a_1, a_1 + a_2, a_1 + a_3), \qquad \mu(b_1, b_2, b_3) = (b_1, b_2 - b_1, b_3 - b_1).$$

## Möbius inversion formula

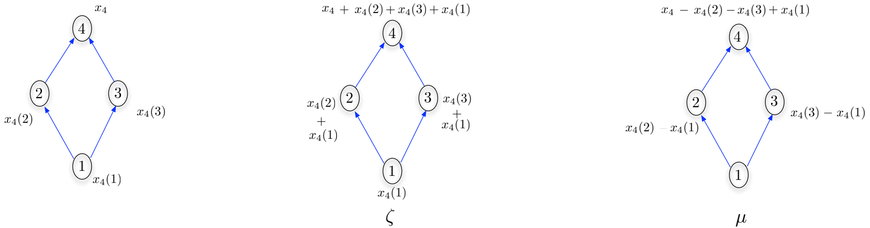
$$g(y) = \sum_{x \preceq y} h(x) \qquad \Leftrightarrow \qquad h(y) = \sum_{x \preceq y} \mu(x,y) g(x)$$

# Möbius inversion: examples

- If $\mathcal{P}$ is a chain: then $\zeta$ is "integration", $\mu := \zeta^{-1}$ is "differentiation".
- If $\mathcal{P}$ is the subset lattice, then $\mu$ is inclusion-exclusion
- If $\mathcal{P}$ is the divisibility integer lattice, then $\mu$ is the number-theoretic Möbius function.
- Many others: vector spaces, faces of polytopes, graphs/circuits, ...

# Möbius inversion is local

- Key insight: Möbius inversion respects the poset structure.
- No additional communication requirements to compute them.
- Thus, can view as operators on local variables: $\zeta(X), \mu(X)$.




$$\zeta(X) = \begin{bmatrix} x_1 & * & * & * \\ x_2(1) & x_2 + x_2(1) & * & * \\ x_3(1) & * & x_3 + x_3(1) & * \\ x_4(1) & x_4(2) + x_4(1) & x_4(3) + x_4(1) & x_4 + x_4(1) + x_4(2) + x_4(3) \end{bmatrix} \quad \mu(X) = \begin{bmatrix} x_1 & * & * & * \\ x_2(1) & x_2 - x_2(1) & * & * \\ x_3(1) & * & x_3 - x_3(1) & * \\ x_4(1) & x_4(2) - x_4(1) & x_4(3) - x_4(1) & x_4 + x_4(1) - x_4(2) - x_4(3) \end{bmatrix}$$

# Controller Architecture

- Let the system dynamics be $x[t+1] = Ax[t] + Bu[t]$, where $A, B \in \mathcal{I}(\mathcal{P})$
- Define controller state variables $X_{ij}$ for $j \preceq i$, where $X_{ii} = x_i$.
- Propose a control law:

$$U = \zeta(\mathbf{G} \circ \mu(X)).$$

  where $\mathbf{G} = \{G(1), \ldots, G(s)\}$.
- Can compactly write closed-loop dynamics as matrix equations:

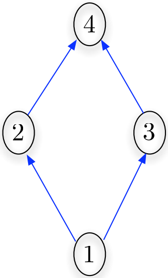
$$X[t+1] = AX[t] + B\zeta(\mathbf{G} \circ \mu(X[t])).$$

  - Each column corresponds to a different subsystem
  - Equations have structure of $\mathcal{I}$, only need entries with $j \preceq i$
  - Diagonal is the plant, off-diagonal is the controller
  - Since $\zeta$ and $\mu$ are local, so is the closed-loop

# Controller Architecture: $U = \zeta(\mathbf{G} \circ \mu(X))$

- "Local errors" computed by $\mu(X)$ (differentiation)
- Compute "local corrections"
- Aggregate them via $\zeta(\cdot)$ (integration)

$$\begin{bmatrix} * \\ u_2 \\ * \\ u_4(2) \end{bmatrix} = G(1) \begin{bmatrix} x_1 \\ x_2(1) \\ x_3(1) \\ x_4(1) \end{bmatrix} + G(2) \begin{bmatrix} * \\ x_2 - x_2(1) \\ * \\ x_4(2) - x_4(1) \end{bmatrix}$$




$$\begin{bmatrix} u_1 \\ u_2(1) \\ u_3(1) \\ u_4(1) \end{bmatrix} = G(1) \begin{bmatrix} x_1 \\ x_2(1) \\ x_3(1) \\ x_4(1) \end{bmatrix}$$

# Separation Principle

- Closed-loop equations:

$$X[t+1] = AX[t] + B\zeta(\mathbf{G} \circ \mu(X[t])).$$

- Apply $\mu$, and use the fact that $\mu$ and $\zeta$ are inverses:

$$\begin{aligned}\mu(X)[t+1] &= A\mu(X)[t] + B(\mathbf{G} \circ \mu(X)[t]) \\ &= (\mathbf{A} + \mathbf{BG}) \circ \mu(X).\end{aligned}$$

  where $(\mathbf{A} + \mathbf{BG})_i = A(\uparrow i, \uparrow i) + B(\uparrow i, \uparrow i)G(i)$.

- "Innovation" dynamics at subsystems decoupled!
- Stabilization easy: simply pick $G(i)$ to stabilize $A(\uparrow i, \uparrow i), B(\uparrow i, \uparrow i)$.

# Separation Principle

- Closed-loop equations:

$$X[t+1] = AX[t] + B\zeta(\mathbf{G} \circ \mu(X[t])).$$

- Apply $\mu$, and use the fact that $\mu$ and $\zeta$ are inverses:

$$\begin{aligned} \mu(X)[t+1] &= A\mu(X)[t] + B(\mathbf{G} \circ \mu(X)[t]) \\ &= (\mathbf{A} + \mathbf{BG}) \circ \mu(X). \end{aligned}$$

where $(\mathbf{A} + \mathbf{BG})_i = A(\uparrow i, \uparrow i) + B(\uparrow i, \uparrow i)G(i)$.

- "Innovation" dynamics at subsystems decoupled!
- Stabilization easy: simply pick $G(i)$ to stabilize $A(\uparrow i, \uparrow i)$, $B(\uparrow i, \uparrow i)$.

# Separation Principle

- Closed-loop equations:

$$X[t+1] = AX[t] + B\zeta(\mathbf{G} \circ \mu(X[t])).$$

- Apply $\mu$, and use the fact that $\mu$ and $\zeta$ are inverses:

$$\begin{aligned} \mu(X)[t+1] &= A\mu(X)[t] + B(\mathbf{G} \circ \mu(X)[t]) \\ &= (\mathbf{A} + \mathbf{BG}) \circ \mu(X). \end{aligned}$$

  where $(\mathbf{A} + \mathbf{BG})_i = A(\uparrow i, \uparrow i) + B(\uparrow i, \uparrow i)G(i)$.

- “Innovation” dynamics at subsystems decoupled!
- Stabilization easy: simply pick $G(i)$ to stabilize $A(\uparrow i, \uparrow i), B(\uparrow i, \uparrow i)$.

# Optimality

**Theorem (Shah-P., CDC2010)**

*$\mathcal{H}_2$-optimal controllers have the described architecture.*

- Gains $G(i)$ obtained by solving decoupled Riccati equations.
- States in the controller are precisely predictions $X_{ij}$ for $j \prec i$.
- Controller order is number of intervals in the poset.

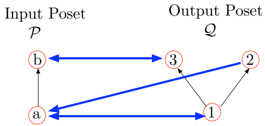
# Controller architecture

## Möbius-inversion controller

$$U = \zeta(\mathbf{G} \circ \mu(X)).$$

Simple and natural structure, for any locally finite poset.

- Can exploit further restrictions (e.g., distributive lattices)
- For product posets, well-understood composition rules for $\mu$
- Generalizes many concepts (Youla parameterization, “purified outputs”, etc)
- Extensions to output feedback, different plant/controller posets (Galois connections), . . .



# Conclusions

- Posets provide useful framework to reason about decentralized decision-making on causal or hierarchical structures.
  - Conceptually nice, computationally tractable.
  - Simple controller structure, based on Möbius inversion.
  - $\mathcal{H}_2$-optimal controllers have this structure.
- Want to know more? $\rightarrow$ www.mit.edu/˜pari
  - "A partial order approach to decentralized control", CDC 2008.
  - "$\mathcal{H}_2$-optimal decentralized control over posets: a state-space solution for state-feedback", CDC 2010.
  - "An optimal architecture for decentralized control over posets", online soon!

Thanks for your attention!

# Conclusions

- Posets provide useful framework to reason about decentralized decision-making on causal or hierarchical structures.
  - Conceptually nice, computationally tractable.
  - Simple controller structure, based on Möbius inversion.
  - $\mathcal{H}_2$-optimal controllers have this structure.
- Want to know more? $\rightarrow$ www.mit.edu/˜pari
  - “A partial order approach to decentralized control”, CDC 2008.
  - “$\mathcal{H}_2$-optimal decentralized control over posets: a state-space solution for state-feedback”, CDC 2010.
  - “An optimal architecture for decentralized control over posets”, online soon!

Thanks for your attention!