NCS Lecture 12
Observability of Guarded Command Programs

Domitilla Del Vecchio (U. Mich) **Richard M. Murray (Caltech)**
20 March 2008

Goals:

- Describe how to design observers for gaurded command programs
- Introduce the use of lattice theory as a tool for analyzing stability

Reading:

- D. Del Vecchio, R. M. Murray, and E. Klavins, “Discrete State Estimators for Systems on a Lattice”, *Automatica*, vol. 42, pp. 271-285, 2006.

Observation of CCL Programs

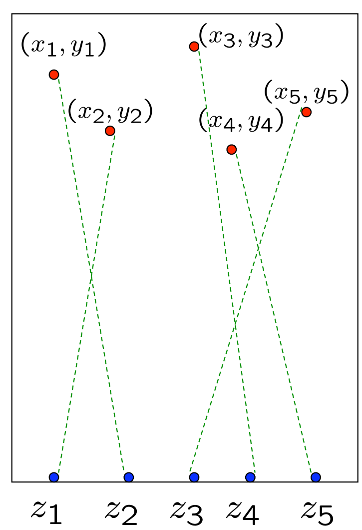
Del Vecchio, M & Klavins, Automatica 2006

Problem: Determine state of communications protocol used by a group of robots given their physical movements.

Assumptions: Protocol and motion control are described in CCL like language.

Results:

- Defns of observability, etc. for CCL programs
- Construction and analysis of observer that converges when the system is "weakly" observable
- Construction of an <u>efficient</u> observer for Roboflag drill in particular
- Everything specified in CCL

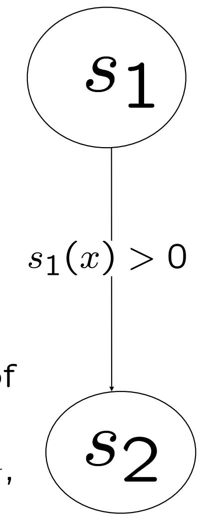


Guarded Command Program

V set of *variable symbols*, with values in U
A state s is a function from V to U
Let S denote the set of states

A **guarded command** is $g : r$ where:
g is a **guard** = a predicate on states
r is a **rule** = a relation on states:

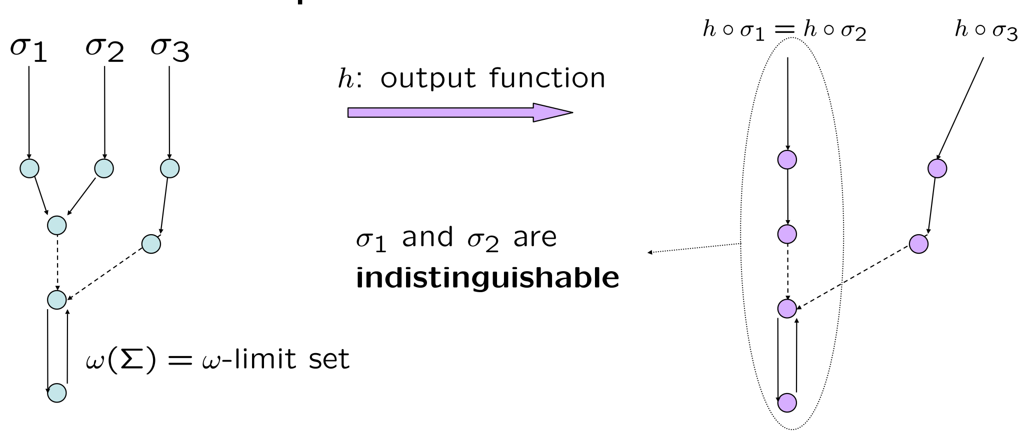
$$s_1(x) > 0 \; : \; s_2(x) = s_1(x) + s_1(y).$$



A **guarded command program** is a set Σ of guarded commands.
An **execution** of Σ is a sequence $\sigma = \{s_t\}_{t \in N}$, such that

$$\forall t \in N \; \exists \, g : r \in \Sigma . g(s) \rightarrow s_t \; r \; s_{t+1}$$

Properties of executions

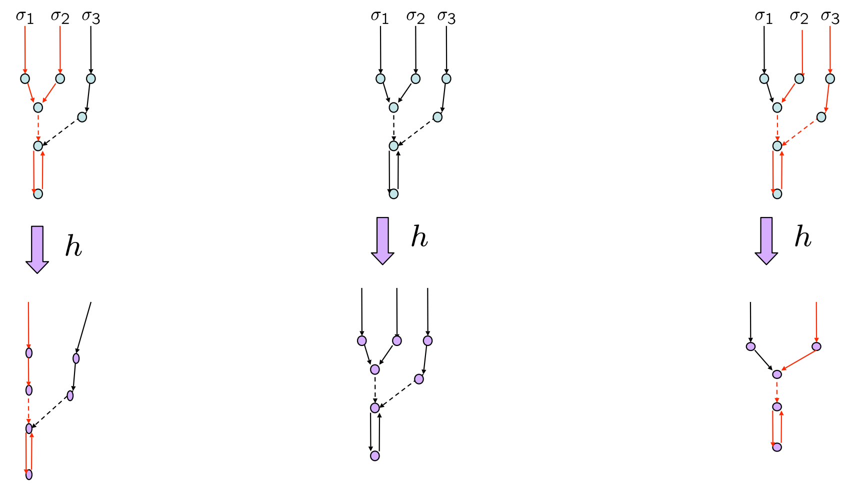


$\sigma_1 \sim \sigma_2$: weakly equivalent executions
if there exists a time t^* such that $\sigma_1(t^*) \notin \omega(\Sigma)$ and $\sigma_1(t) = \sigma_2(t)$ for all $t \geq t^*$.

σ_2, σ_3: non-equivalent executions

Observability

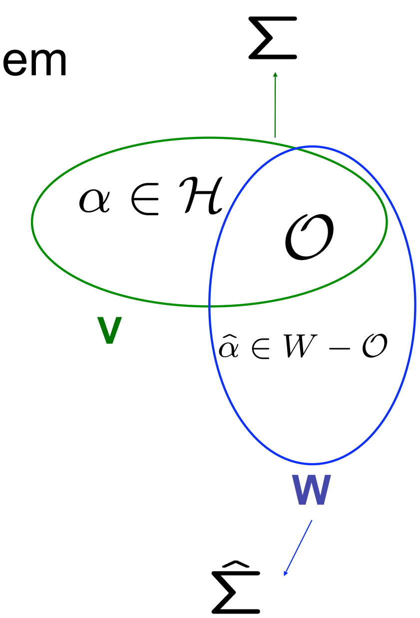
Weakly observable — Observable — Non-observable



The system is **weakly observable** if whenever σ_1 and σ_2 are not equivalent then σ_1 and σ_2 are distinguishable.

The guarded command program Σ is said to be **observable** with respect to the output function $h : S \to U$ if any two executions $\sigma_1, \sigma_2 \in \mathcal{E}(\Sigma)$ are distinguishable.

Observer Problem



Let $V = \mathcal{H} \cup \mathcal{O}$ and W be such that $\mathcal{O} \subseteq W$ and $\mathcal{H} \cap W = \emptyset$. Suppose that α is the vector of all variables in $\mathcal{H}$ and suppose that $\hat{\alpha} \in W - \mathcal{O}$. Given a guarded command program Σ, the guarded command program $\hat{\Sigma}$ is an **observer** for Σ if the following hold for all $\sigma \in \mathcal{E}(\Sigma \bigcup \hat{\Sigma})$:

(i) there exists a time t^* such that $\hat{\alpha}(t) = \alpha(t)$ for all $t \geq t^*$;

(ii) there exists a metric d on type($\hat{\alpha}$) such that for each ε there exists a δ such that for all t

$$d(\hat{\alpha}(0), \alpha(0)) < \delta \;\Rightarrow\; d(\hat{\alpha}(t), \alpha(t)) < \varepsilon.$$

The Model

Let $\mathcal{O} = \{z_1, ..., z_{N_\mathcal{O}}\}$ and $\mathcal{H} = \{\alpha_1, ..., \alpha_{N_\mathcal{H}}\}$ and put $V = \mathcal{O} \cup \mathcal{H}$.

$$\Sigma \quad \begin{cases} P_{i,j}(z,\alpha) : z_i' = f_{i,j}(z), & j \in \{1, ..., K_i\} \\ Q_{k,l}(z,\alpha) : \alpha_k' = g_{k,l}(\alpha) & l \in \{1, ..., M_k\}, \end{cases}$$

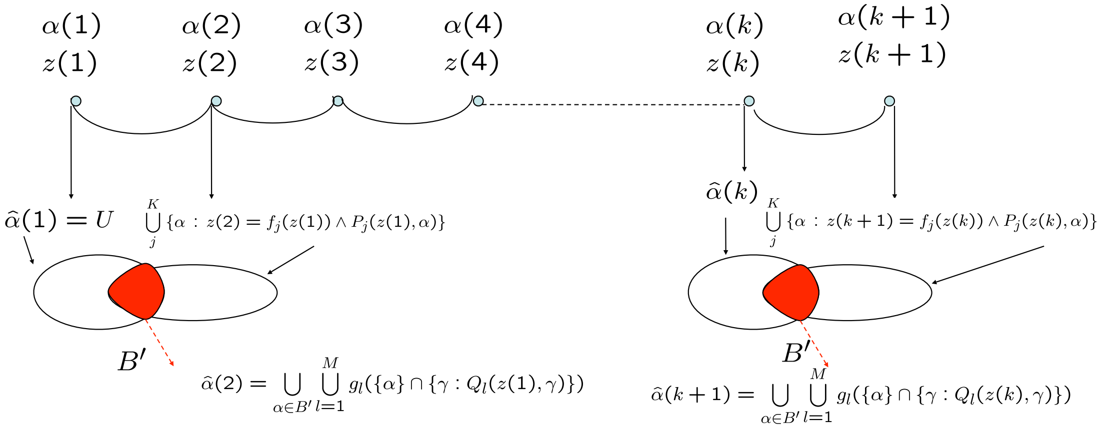
(A1) For each i there is exactly one $j \in \{1, ..., K_i\}$ such that $P_{i,j}(z,\alpha)$ is true, and for each k there is exactly one $l \in \{1, ..., M_k\}$ such that $Q_{k,l}(z,\alpha)$ is true.

Observer Construction

Initially $\hat{\alpha} = U$

$$\hat{\Sigma} \quad \begin{cases} true : B' = \bigcap_i^{N_\mathcal{O}} \bigcup_j^{K_i} \{\alpha \, : \, z_i' = f_{i,j}(z) \wedge P_{i,j}(z,\alpha)\} \cap \hat{\alpha} \\ \wedge \ \hat{\alpha}' = \bigcup_{\alpha \in B'} \left\{ \beta : \forall k. \beta_k \in \bigcup_{l=1}^{M_k} g_{k,l}(\{\alpha\} \cap \{\gamma : Q_{k,l}(z,\gamma)\}) \right\} \end{cases}$$

Schematic with i=1



Is it an observer?

Theorem: Given Σ, the program $\hat{\Sigma}$ satisfies the following properties:

(1) For all t, $\alpha(t) \in \hat{\alpha}(t)$ (correctness);

(2) If Σ is weakly observable,then $\hat{\alpha}$ converges to α; (convergence)

(3) there exists a δ such that for all t

$$d(\hat{\alpha}(0), \alpha(0)) < \delta \;\Rightarrow\; d(\hat{\alpha}(t), \alpha(t)) < \varepsilon.$$

(small error).

Therefore, $\hat{\Sigma}$ is an observer for Σ.

Example: RoboFlag Drill

Red robots $\quad \Sigma_{Red} \quad y_i - \delta > 0 \;:\; y_i' = y_i - \delta$

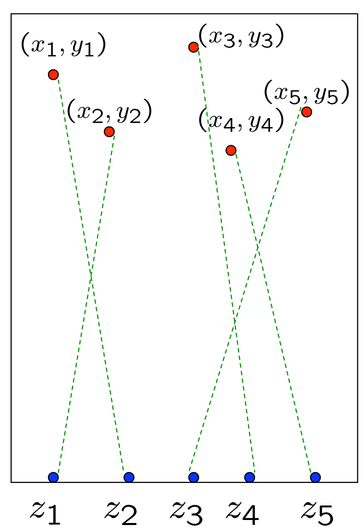
Blue robots

$$\Sigma_{Blue} \begin{cases} z_i < x_{\alpha_i} & : \; z_i' = z_i + \delta \\ z_i > x_{\alpha_i} & : \; z_i' = z_i - \delta \\ z_i = x_{\alpha_i} & : \; z_i' = z_i \end{cases}$$

Assignment

$$\Sigma_{Assign} \begin{cases} i = \{2, ..., N-1\} \\ down_i \; := \; up_{i-1} \\ up_i \; := \; \neg down_i \wedge x_{\alpha_i} > x_{\alpha_{i+1}} \\ \\ down_1 \; := \; false \qquad down_N \; := \; up_{N-1} \\ up_1 \; := \; x_{\alpha_1} > x_{\alpha_2} \qquad up_N \; := \; false\;. \\ \\ down_i \; : \; \alpha_i' = \alpha_{i-1} \\ up_i \; : \; \alpha_i' = \alpha_{i+1} \\ \neg(down_i \vee up_i) \; : \; \alpha_i' = \alpha_i. \end{cases}$$

$$\Sigma_{RF} := \Sigma_{Red} \cup \Sigma_{Blue} \cup \Sigma_{Assign}.$$



$\alpha = [2, 1, 5, 3, 4]$

Observation Problem

Estimation of the blue robot assignment, given the observation of the of the z variables, and knowing the x and y variables (treated as parameters)

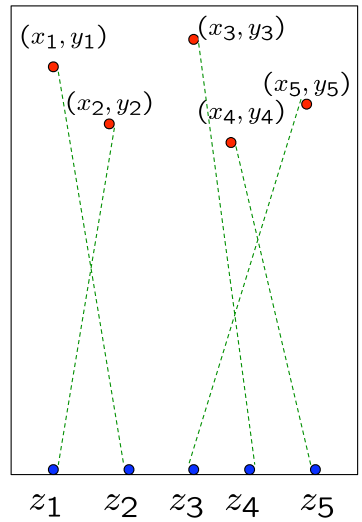
$$\Sigma = \Sigma_{Blue} \cup \Sigma_{Assign}$$

$$\mathcal{O} = \{z_1, ..., z_N\} \qquad \mathcal{H} = \{\alpha_1, ..., \alpha_N\}$$

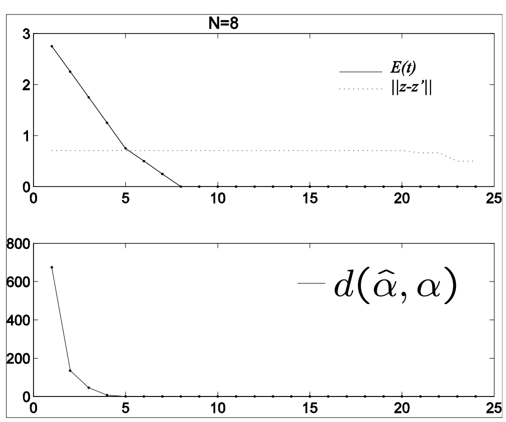
Assumption: $x_i \in (z_{i-1}, z_i)$ during the entire execution

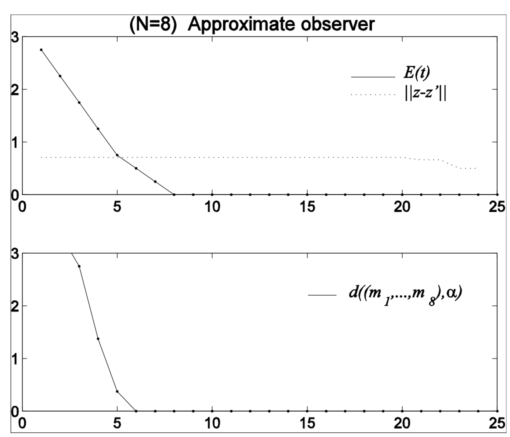
Lemma: The program $\Sigma_{Blue} \cup \Sigma_{Assign}$ is weakly observable

Proposition: The observer $\hat{\Sigma}$ applied to $\Sigma_{Blue} \cup \Sigma_{Assign}$ converges in at most $t_\sigma^\alpha + 1$ steps in any execution σ of $\Sigma_{Blue} \cup \Sigma_{Assign} \cup \hat{\Sigma}$.



Simulation Results

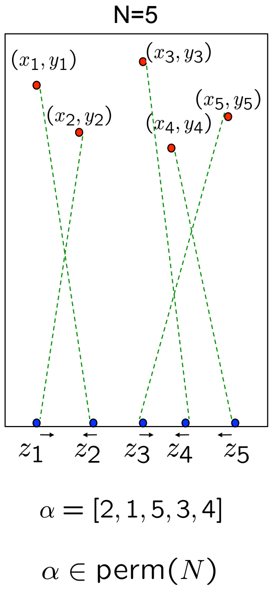




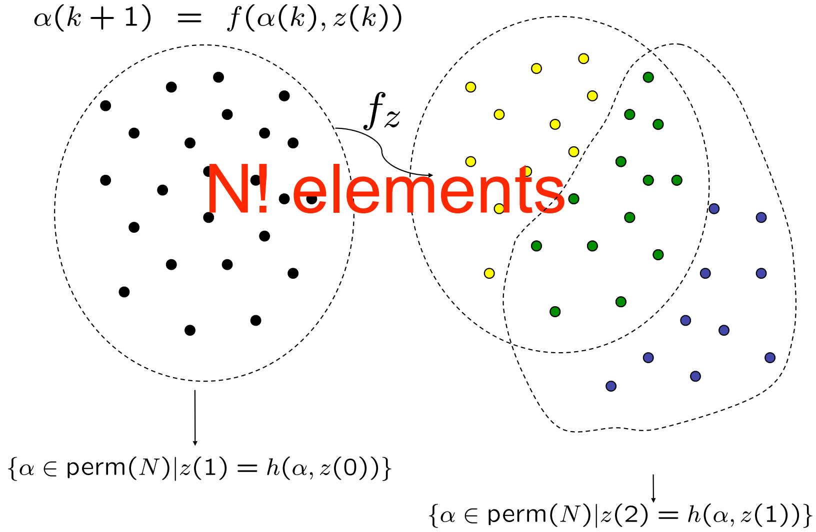
$$E(t) = \frac{1}{N} \sum_{i=1}^{N} |\alpha_i - i|,$$

$$d((m_1, ..., m_N), \alpha) := \frac{1}{N} \sum_{i=1}^{N} d(m_i, \alpha_i)$$

The location observation tree methods lead to combinatorial complexity



$$z(k+1) = h(\alpha(k), z(k))$$
$$\alpha(k+1) = f(\alpha(k), z(k))$$

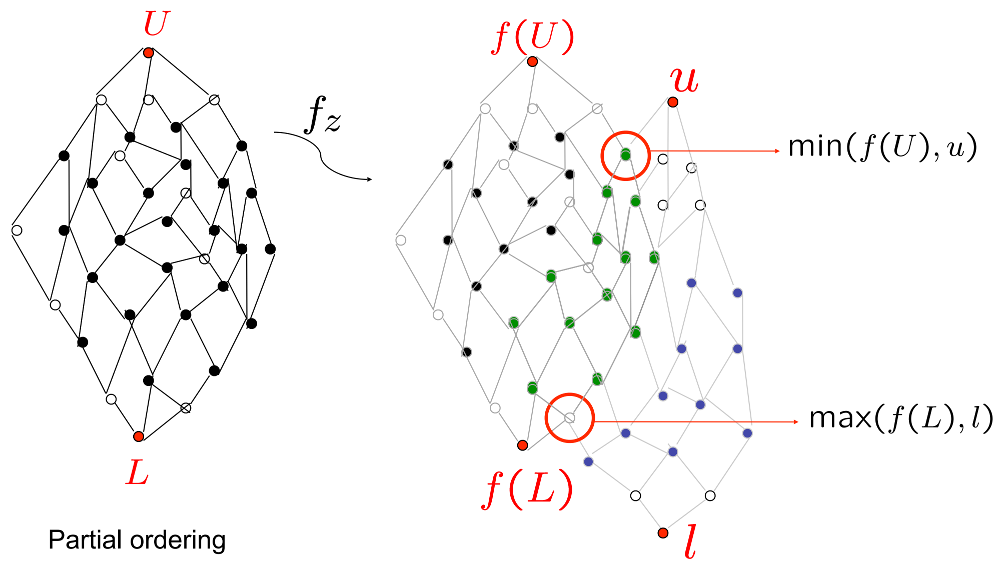


We need a low computation current state estimator

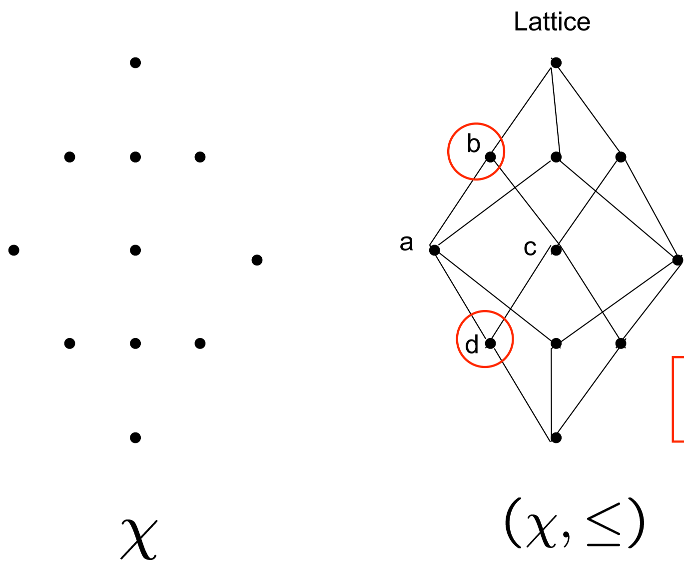
- Low means comparable to the computation needed for simulating the system under study itself

- We look for the “cheapest” representation of a set :
  the list-representation of a set is the most expensive!

Complexity can be reduced by superimposing a partial order



A partial ordering allows to represent a set by two elements only



a<b

a not rel c

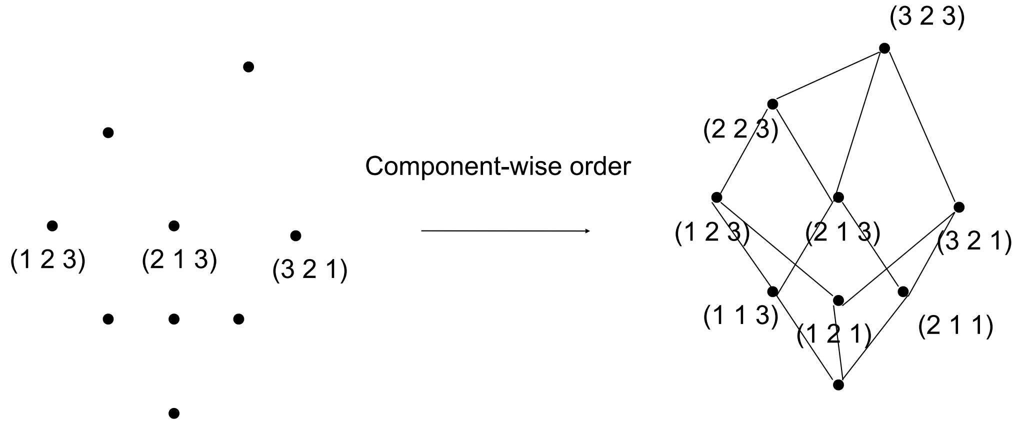
max(a,c) := a ∨ c = b

min(a,c) := a ∧ c = d

{a,b,c,d}=[d,b]

Example: vectors with natural entries

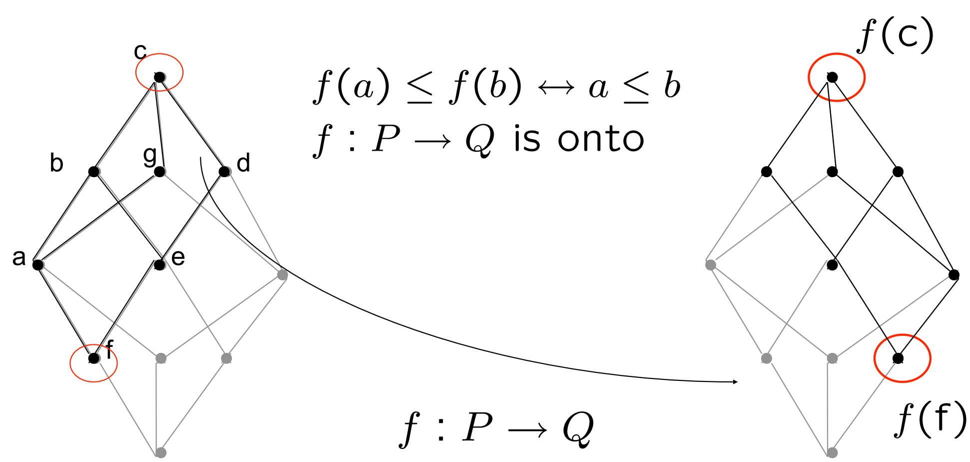
$\alpha \in \mathbb{N}^3$ with entries less than 3



Order isomorphic maps preserve the lattice structure

$f(a) \leq f(b) \leftrightarrow a \leq b$

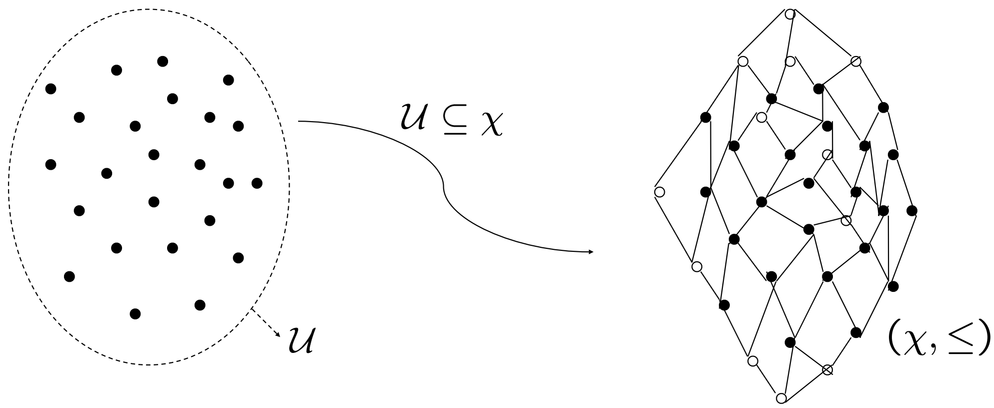
$f : P \rightarrow Q$ is onto



{a,b,c,d,e,f,g}=[f,c]

f([f,c])=[f(f),f(c)]

A system can always be extended to a lattice



$$\Sigma = (f, h, \mathcal{U}, \mathcal{Z})$$

$$\begin{aligned} \alpha(k+1) &= f(\alpha(k), z(k)) \\ z(k+1) &= h(\alpha(k), z(k)) \end{aligned}$$

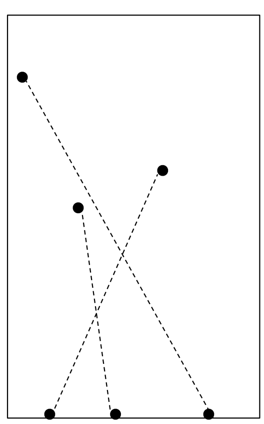
$\alpha \in \mathcal{U}$
$z \in \mathcal{Z}$

$$\tilde{\Sigma} = (\tilde{f}, \tilde{h}, \chi, \mathcal{Z})$$

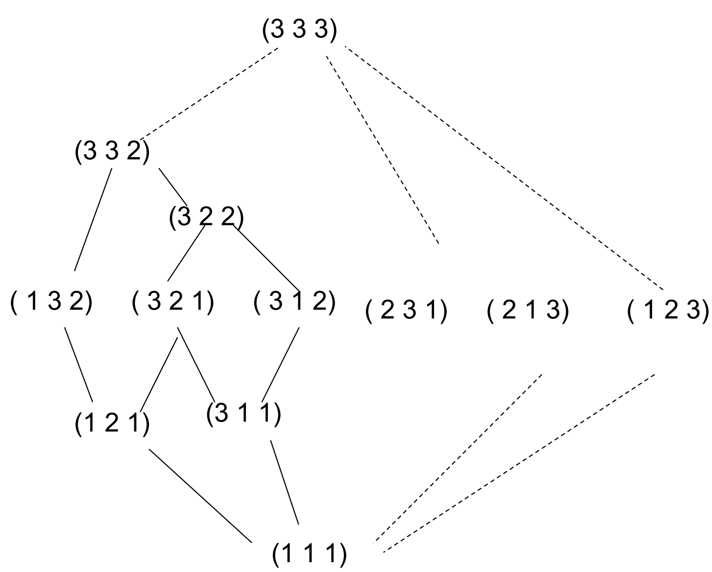
$$\begin{aligned} w(k+1) &= \tilde{f}(w(k), z(k)) \\ z(k+1) &= \tilde{h}(w(k), z(k)) \end{aligned}$$

$w \in \chi$ $\quad \tilde{f}|_{\mathcal{U}} = f$
$z \in \mathcal{Z}$ $\quad \tilde{h}|_{\mathcal{U}} = h$

Example: set of permutations extended to the set of vectors with natural entries



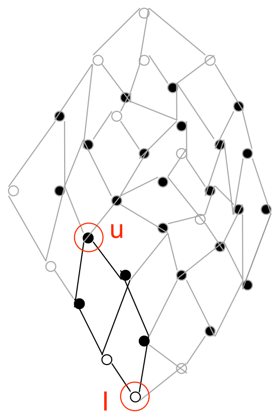
$\alpha \in \text{perm}(3)$
$\alpha = (3\ 2\ 1)$



Order compatible pairs allow to reduce complexity

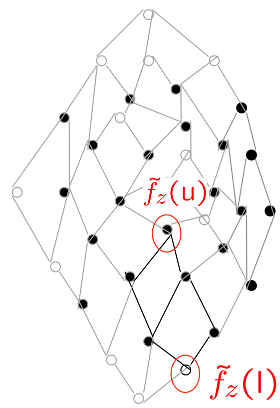
$\tilde{\Sigma} = (\tilde{f}, \tilde{h}, \chi, \mathcal{Z})$ and $(\chi, \leq)$ are order compatible if

1) The output set is an interval



$\{w \in \chi \mid z(k+1) = \tilde{h}(w, z(k))\} =$[l,u]

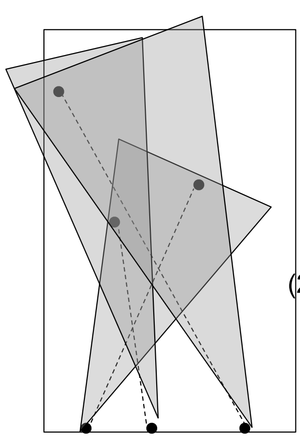
2) $\tilde{f}_z$ Is order isomorphic on the output set



$\tilde{f}_z$([l,u])=[$\tilde{f}_z$(l),$\tilde{f}_z$(u)]

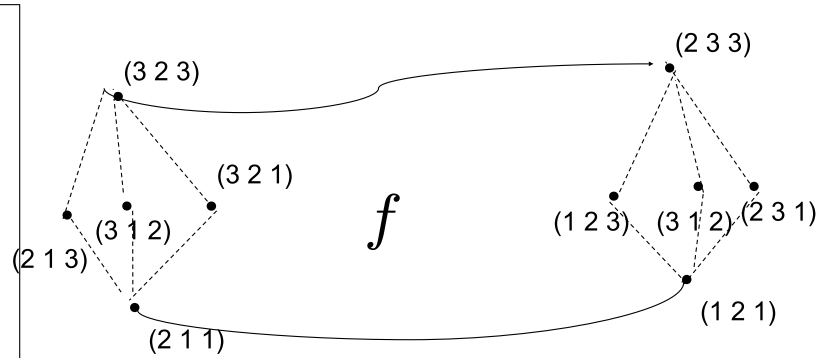
Example: RoboFlag Drill

$\alpha \in \mathbb{N}^3$ with entries less than 3



$\alpha \in$ perm(3)

$\alpha = (3\ 2\ 1)$

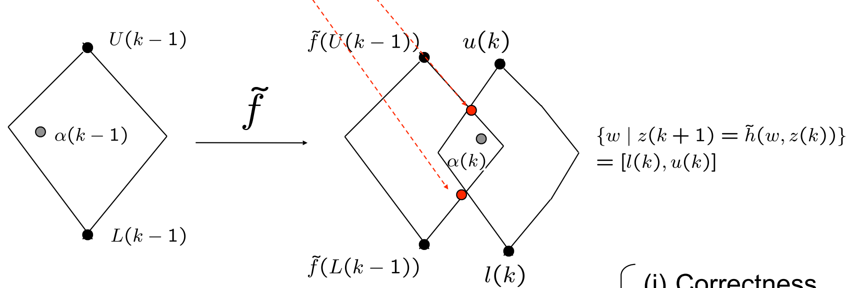


$\alpha_i > i \wedge \alpha_{i+1} \leq i+1 \;:\; (\alpha'_i, \alpha'_{i+1}) = (\alpha_{i+1}, \alpha_i)$

Theorem: The discrete state estimation problem for order compatible pairs is solved by updating two variables

$$U(k) = \min\{\tilde{f}(U(k-1)), u(k)\}$$
$$L(k) = \max\{\tilde{f}(L(k-1)), l(k)\}$$

$$U(0) = \sup(\chi) \text{ and } L(0) = \inf(\chi)$$

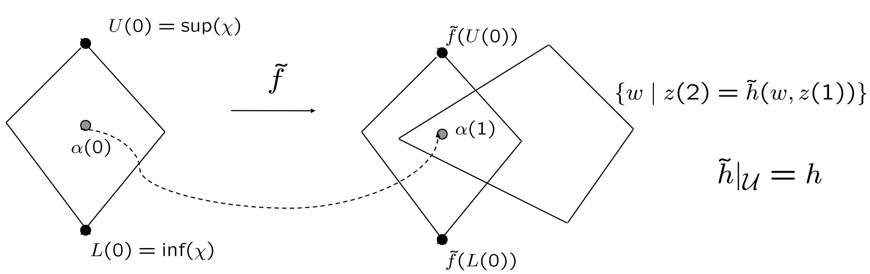


For these updates we have three properties:

- (i) Correctness
- (ii) Non-increasing error
- (iii) Convergence

Estimator property (i): correctness

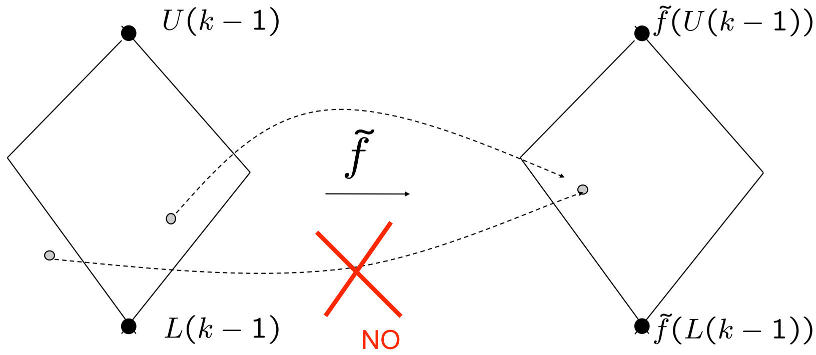
$$L(k) \leq \alpha(k) \leq U(k) \text{ for all } k$$



$$\tilde{h}|_{\mathcal{U}} = h$$

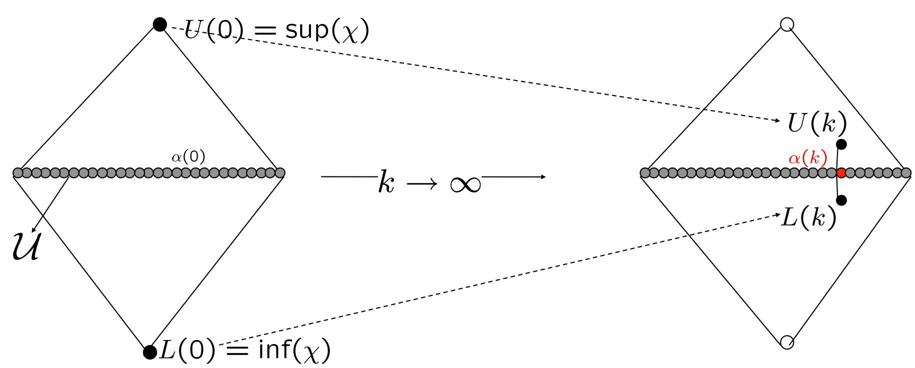
Estimator property (ii): non-increasing error

$|[L(k), U(k)]| \leq |[L(k-1), U(k-1)]|$



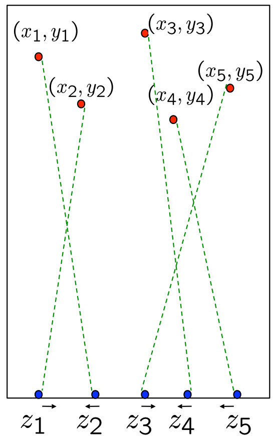
Estimator property (iii): convergence

If Σ is observable then $[L(k), U(k)] \cap \mathcal{U} \to \alpha(k)$



If $\tilde{\Sigma}$ is observable then $L(k) \to U(k) \to \alpha(k)$

RoboFlag example: the set of permutations is extended to the set of vectors with natural entries



$\alpha = [2, 1, 5, 3, 4]$

$\alpha \in \text{perm}(N)$

$$z' = h(\alpha, z) \begin{cases} z_i < x_{\alpha_i} & : \; z'_i = z_i + \delta \\ z_i > x_{\alpha_i} & : \; z'_i = z_i - \delta \\ z_i = x_{\alpha_i} & : \; z'_i = z_i \end{cases}$$

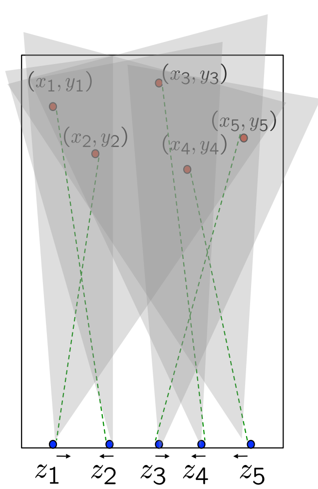
$$\alpha' = f(\alpha, z) \quad x_{\alpha_i} \geq z_{i+1} \wedge x_{\alpha_{i+1}} \leq z_{i+1} \;:\; (\alpha'_i, \alpha'_{i+1}) = (\alpha_{i+1}, \alpha_i)$$

perm(N) extended to $\mathbb{N}^N = \chi$
componentwise order

$$z' = \tilde{h}(w, z) \begin{cases} z_i < x_{w_i} & : \; z'_i = z_i + \delta \\ z_i > x_{w_i} & : \; z'_i = z_i - \delta \\ z_i = x_{w_i} & : \; z'_i = z_i \end{cases} \qquad w \in \mathbb{N}^N$$

$$w' = \tilde{f}(w, z) \quad x_{w_i} \geq z_{i+1} \wedge x_{w_{i+1}} \leq z_{i+1} \;:\; (w'_i, w'_{i+1}) = (w_{i+1}, w_i)$$

RoboFlag example: The extended system and the lattice $(\mathbb{N}^N, \leq)$ are order compatible



$\alpha = [2, 1, 5, 3, 4]$

$\alpha \in \text{perm}(N)$

1) The output set is an interval

$$\{w \in \mathbb{N}^N \mid z(k+1) = \tilde{h}(w, z(k))\} = [\begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 5 \\ 2 \\ 5 \\ 4 \\ 5 \end{pmatrix}]$$

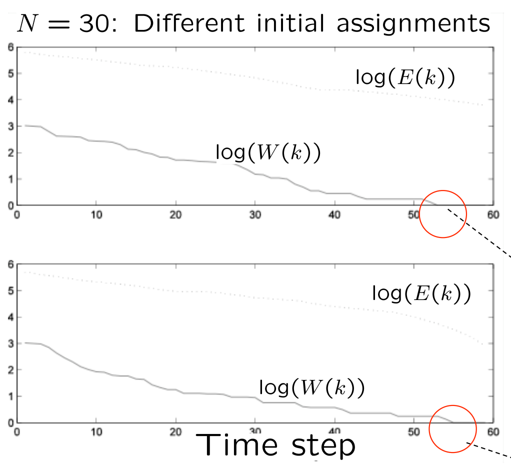
2) $\tilde{f}$ is order isomorphic: $\tilde{f}([l, u]) = [\tilde{f}(l), \tilde{f}(u)]$

$$\tilde{f}([\begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 5 \\ 2 \\ 5 \\ 4 \\ 5 \end{pmatrix}]) = [\tilde{f}(\begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \\ 1 \end{pmatrix}), \tilde{f}(\begin{pmatrix} 5 \\ 2 \\ 5 \\ 4 \\ 5 \end{pmatrix})] = [\begin{pmatrix} 1 \\ 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 5 \\ 4 \\ 5 \\ 5 \end{pmatrix}]$$

The complexity of the RoboFlag estimator is about 2*complexity of the RoboFlag system

- We have 2N variables
- We have 2N clauses for updating L and U
- W have 2N computations of max and min between natural numbers

Simulation results: the estimator can be run in systems with large N



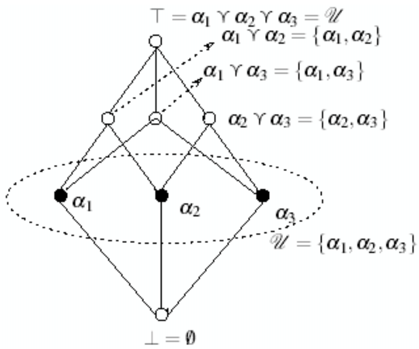
$$E(k) = \frac{1}{N}\sum_{i=1}^{N} |\alpha_i(k) - i|$$

$$W(k) \propto |[L(k), U(k)] \cap \mathcal{U}|$$

Estimator converged

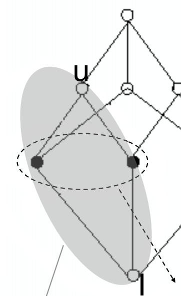
The proposed estimator exists for any observable system

$(\chi, \leq) := (\mathcal{P}(\mathcal{U}), \subseteq)$



For $x, w \in \chi$

$$\tilde{f}(x \vee w) := \tilde{f}(x) \vee \tilde{f}(w)$$
$$\tilde{f}(x \wedge w) := \tilde{f}(x) \wedge \tilde{f}(w)$$



$\{\alpha \in \mathcal{U} \mid z(k+1) = h(\alpha, z(k))\}$

$\{w \in \chi \mid z(k+1) = \tilde{h}(w, z(k))\}$=[l,u]

$(\chi, \leq)$ and $\tilde{\Sigma} = (\tilde{f}, \tilde{h}, \chi, \mathcal{Z})$ are order compatible

The computational burden never exceeds the one of the observer tree method

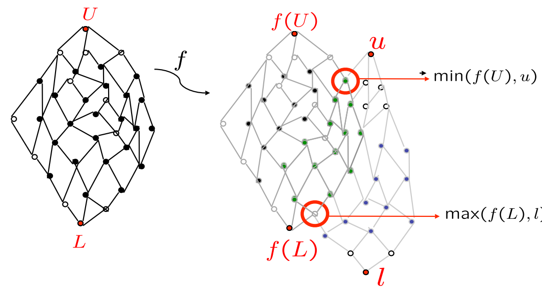
- The observer tree requires $O(|\mathcal{U}|^2)$ computations (Caines 1991)
- The size of the lattice $(\chi, \leq)$ is less than $2|\mathcal{U}|^2$

- The lattice approach to estimation has a worst-case computational burden equivalent to previously proposed methods

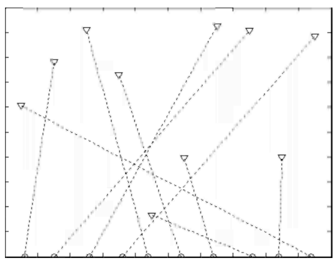
- When the system has a preferred partial order structure the computation can be drastically reduced and scalability in the number of variables to be estimated can be reached

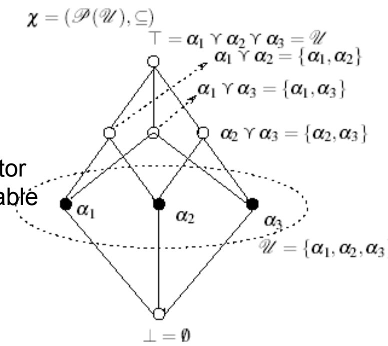
Conclusions

The problem of estimating the discrete state in hybrid systems can be computationally intractable if “cheap” sets representations are not employed



The estimator has been applied to a multi-robot system with large numbers of agents





The proposed estimator exists for any observable system