State Estimation and Event Inference in DES

Applications to Fault Diagnosis and Opacity Verification

Christoforos N. Hadjicostis

Department of Electrical and Computer Engineering
University of Cyprus

Acknowledgements
DES Collaborators and Funding

- Doctoral students (former and current): E. Athanasopoulou, C. Keroglou, Tung Le, Lingxi Li, M. Panteli, Yu Ru, A. Saboori, Yingquan Wu

- Other colleagues and collaborators: M.-P. Cabasino, C. D. Charalambous, T. Charalambous, A. Giua, E. Fabre, R. Kumar, S. Lafortune, B. Lennartson, F. Lin, J. van Schuppen, C. Seatzu, ...

- Financial support: University of Illinois, National Science Foundation, Air Force Office of Scientific Research, Boeing, Lucent, University of Cyprus, European Commission, Cyprus Research Promotion Foundation, Qatar Foundation, ...

- “Classical" Applications: Manufacturing systems; baggage handling systems; paper handling systems (copiers, printers, etc.); heating, ventilation and air conditioning units

  Main characteristics and challenges: Model-based (human designed), few or unreliable sensors, different modes of operation (e.g., monitoring vs testing), on-line estimation/inference algorithms, off-line verification of properties of interest and complexity analysis

- Emerging Applications: Distributed (cyber-physical) systems, such as autonomous vehicles and automated highway systems; microgrids and smart grids; smart devices and buildings

  Additional characteristics and challenges:
  - Distributivity/Modularity: Multiple systems, observers, controllers
  - Communication: Network delays, packet drops, synchronization
  - Processing: Local vs global, exchange of information
  - Trust/Privacy: Intruders, curious/malicious components, collaborative vs antagonistic strategies

Common DES models:

- Finite automata, both deterministic and nondeterministic
- Petri nets, both bounded and unbounded
- Extensions: Timed models, stochastic models, hybrid models, ...
  Diverse levels of abstraction: Logical, stochastic, hierarchical, ...

Observability related tasks:

1. State estimation and state isolation (e.g., current state estimation)
2. Event inference and categorization (e.g., fault detection/diagnosis)

Challenges discussed here:

1. Online algorithms for state estimation and event inference (centralized)
2. Verification and complexity of properties of interest (centralized)
   (namely, detectability, diagnosability, and opacity)
3. Decentralized/distributed observation, protocol design/verification
4. Automated tools for verifying/enforcing properties of interest

Observability Related Notions in DES
Typical Objectives/Tasks

- Sources of uncertainty
  -Common: initial state, partial event observation, nondeterminism
  -Not-so-common: loss, delay, corruption of observations

- Observability: Detectability and State Isolation
  State (eventually) completely identified or isolated within certain sets
  -Applications: Supervisory control, deadlock avoidance, state avoidance

- Diagnosability: Fault Detection and Fault Diagnosis
  Fault (or other special) events (eventually) detected and/or classified
  -Applications: Monitoring, diagnosis, sensor selection, sensor activation

- Opacity: Security and Privacy-Preservation
  Private system behavior never gets exposed
  -Applications: Privacy analysis and enforcement, security guarantees

- Distinction: Recursive estimation and inference (on-line) versus verification of properties (off-line)

Talk Outline

- Processing of Observations (centralized)
  1. Recursive current state estimation
  2. Generalizations to initial/delayed state estimation
- Properties of Interest and their Verification
  1. Detectability
  2. Fault detection and diagnosis
  3. Current-state and initial-state opacity
- Decentralized Observation Settings (no communication between sites)
  - Fusion of information at coordinator
  - Conveying decisions, or estimates, or observation sequences
  - Verification of properties of interest
- Distributed Observation Settings (some communication between sites)

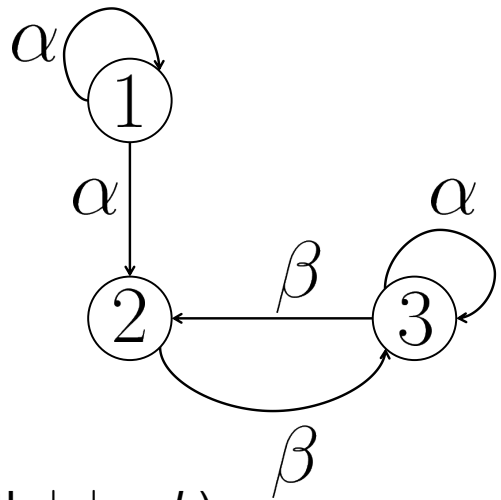
Nondeterministic Finite Automaton (NFA)
Notation

$G = (X, \Sigma, \delta, X_0)$, where

- X is the set of states
- Σ is the set of events
- $\delta : X \times \Sigma \to 2^X$ is the nondeterministic state transition function
  (Deterministic: $|\delta(x, \sigma)| \leq 1$ for all $x \in X$ and $\sigma \in \Sigma$)
- $X_0 \subseteq X$ is the set of possible initial states

Example NFA

$X = \{1, 2, 3\}$ and $\Sigma = \{\alpha, \beta\}$
$X_0 = \{1, 2, 3\}$
For $s = \alpha\beta\beta$, we have
$\delta(\{1, 3\}, s) = \{2, 3\}$



Sequence of events: $s = \sigma_{i_1}\sigma_{i_2}...\sigma_{i_k} \in \Sigma^*$ (of length $|s| = k$)

Behavior of G (language $L(G)$): $L(G) := \{s \in \Sigma^* \mid \exists x_0 \in X_0\{\delta(x_0, s) \neq \emptyset\}\}$

Extended δ function: $\delta(X', \sigma) := \cup_{x' \in X'} \delta(x', \sigma)$ for $X' \subseteq X$, $\sigma \in \Sigma$
$\delta(x, \sigma s) := \delta(\delta(x, \sigma), s)$ for $x \in X$, $s \in \Sigma^*$, $\sigma \in \Sigma$

Post-language (continuations) of $s \in L(G)$: $L(G) \setminus s = \{t \in \Sigma^* : st \in L(G)\}$

Observability Limitations

Unobservable Events (Σ_u) and Observable Events (Σ_o)

- Unobservable events Σ_u, $\Sigma_u \subset \Sigma$: Events whose occurrence goes unrecorded; remaining events $\Sigma_o = \Sigma \setminus \Sigma_u$ are observable

- Natural projection $P_{\Sigma_o} : \Sigma^* \to \Sigma_o^*$ (denoted by P when Σ_o is implied) Defined recursively $\forall \sigma \in \Sigma, s \in \Sigma^*$

$$P(\epsilon) = \epsilon \text{ and } P(\sigma s) = P(\sigma)P(s)$$

where

$$P(\sigma) = \begin{cases} \sigma, & \text{if } \sigma \in \Sigma_o \\ \epsilon, & \text{otherwise} \end{cases}$$

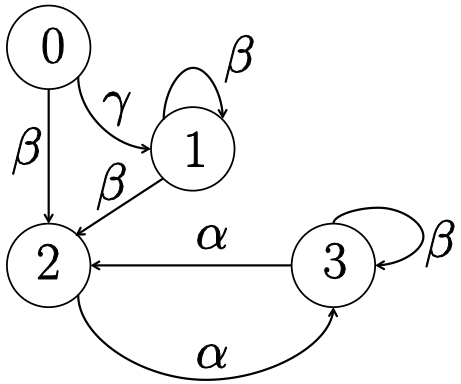
- Natural projection “erases" all unobservable events in s

- Event sequences compatible with sequence of observations $\omega \in \Sigma_o^*$:

$P^{-1}(\omega) = \{s \in L(G) \mid P(s) = \omega\}$ (inverse projection, “explanations")

- Generalizations (not addressed here):
  1. Different events could generate identical observations (“labels")
  2. Additional information (e.g., probabilistic/timing data)

Current State Estimation
Example of Recursive Computation of Possible Current States



- Consider NFA G given above with $X_0 = \{0, 1, 2, 3\}$ and $\Sigma_o = \{\alpha, \beta\}$
- Current state estimation: Given a streaming sequence $\omega \in \Sigma_o^*$, track online possible current states
  $\widehat{X}(\omega) = \{x \in X \mid \exists x_0 \in X_0, \exists s \in \Sigma^* \text{ s.t. } P(s) = \omega \text{ and } x \in \delta(x_0, s)\}$

  E.g., if we observe $\omega = \alpha\beta\alpha$, we can infer $\widehat{X}(\omega) = \{2\}$

  Can recursively track possible current states ($\Rightarrow$ online algorithm)

$$\left\{ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \right\} \xrightarrow{\alpha} \left\{ \begin{array}{c} 2 \\ 3 \end{array} \right\} \xrightarrow{\beta} \{\ 3\ \} \xrightarrow{\alpha} \{\ 2\ \}$$

- Objective: Following an unknown sequence of events $s \in \Sigma^*$, resulting in a sequence of observations $\omega = P(s) \in \Sigma_o^*$, obtain *current* state estimates, i.e.,

$$\widehat{X}(\omega) = \{x \in X \mid \exists x_0 \in X_0, \exists s' \in \Sigma^* \text{ s.t. } P(s') = \omega \text{ and } x \in \delta(x_0, s')\}$$

- Reachable set of states under a single observation: For $X' \subseteq X$, $\sigma_o \in \Sigma_o \cup \{\epsilon\}$, we let the set of states reachable from X' "via observation σ_o" (or "no observation" when $\sigma_o = \epsilon$) be

$$R(X', \sigma_o) = \{x \in X \mid \exists x' \in X', \exists s \in \Sigma^* \text{ s.t. } P(s) = \sigma_o \text{ and } x \in \delta(x', s)\}$$

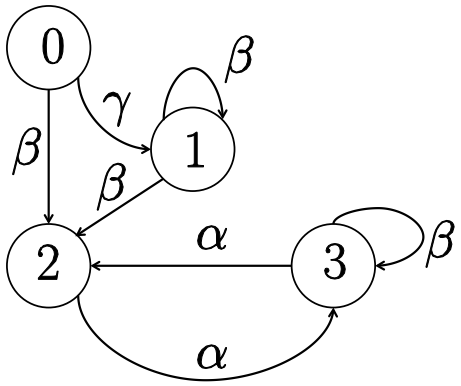
- Given $\omega = \sigma_{i_1}\sigma_{i_2}...\sigma_{i_k} \in \Sigma_o^*$ and $\sigma_{i_{k+1}} \in \Sigma_o$, we can obtain the set $\widehat{X}$ recursively as

$$\begin{aligned} \widehat{X}(\epsilon) &= R(X_0, \epsilon) \text{ "unobservable reach } UR(X_0)\text{"} \\ \widehat{X}(\omega\sigma_{i_{k+1}}) &= R(\widehat{X}(\omega), \sigma_{i_{k+1}}) \end{aligned}$$

- Need: Knowledge of system model or $R(X', \sigma_o)$ for $X' \subseteq X$ and $\sigma_o \in \Sigma_o$
- Note: $\widehat{X}$, P, R, UR, are all defined with respect to Σ_o (should really be $\widehat{X}_{\Sigma_o}$, P_{Σ_o}, R_{Σ_o}, UR_{Σ_o})

Initial State Estimation

Example of Recursive Computation of Possible Initial States



- Again, consider NFA G with $X_0 = \{0, 1, 2, 3\}$ and $\Sigma_o = \{\alpha, \beta\}$
- Initial state estimation: Given a streaming sequence $\omega \in \Sigma_o^*$, track online possible initial states
  $\widehat{X}_0(\omega) = \{x_0 \in X_0 \mid \exists s \in \Sigma^* \text{ s.t. } P(s) = \omega \text{ and } \delta(x_0, s) \neq \emptyset\}$
- Key idea: Track possible pairs (x_i, x_c) of an initial state $x_i \in X_0$ and a *matching* current state $x_c \in X$
- For example, if we observe $\omega = \alpha\beta\alpha$, we can recursively obtain

$$\left\{ \begin{array}{c} (0,0) \\ (0,1) \\ (1,1) \\ (2,2) \\ (3,3) \end{array} \right\} \xrightarrow{\alpha} \left\{ \begin{array}{c} (3,2) \\ (2,3) \end{array} \right\} \xrightarrow{\beta} \{ \ (2,3) \ \} \xrightarrow{\alpha} \{ \ (2,2) \ \}$$

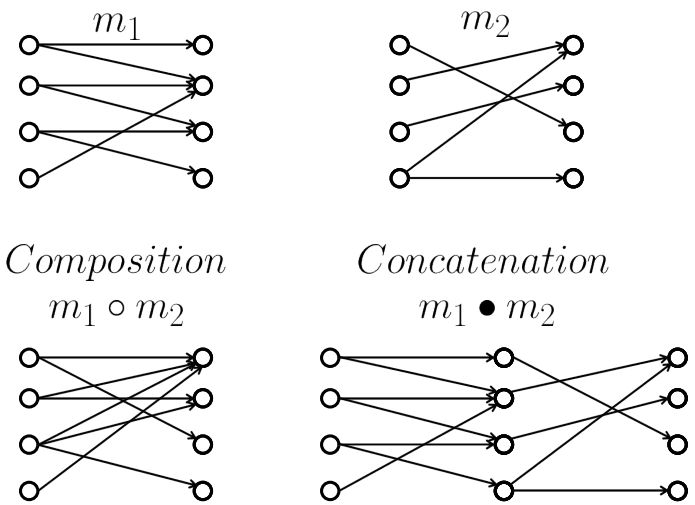
Recursive State Estimation

State Mappings, Composition, Concatenation [Hadjicostis, 2018]

- A state mapping $m \subseteq X \times X$ contains pairs of the form (x_{i_1}, x_{i_2}) where x_{i_1} (x_{i_2}) can be thought as current (next) state
- State mappings $m_1, m_2 \subseteq X \times X$ can be composed (to generate a new state mapping) or concatenated (to generate a trellis diagram) as

$$\begin{aligned} m_1 \circ m_2 &= \{(x_{i_1}, x_{i_3}) \mid \exists x_{i_2} \in X \text{ s.t. } (x_{i_1}, x_{i_2}) \in m_1 \text{ and } (x_{i_2}, x_{i_3}) \in m_2\} \\ m_1 \bullet m_2 &= \{(x_{i_1}, x_{i_2}, x_{i_3}) \mid \exists (x_{i_1}, x_{i_2}) \in m_1, (x_{i_2}, x_{i_3}) \in m_2\} \end{aligned}$$

- Graphical depiction of composition and concatentation (arrows connect a state $x_{i_1} \in X$ with another state $x_{i_2} \in X$)

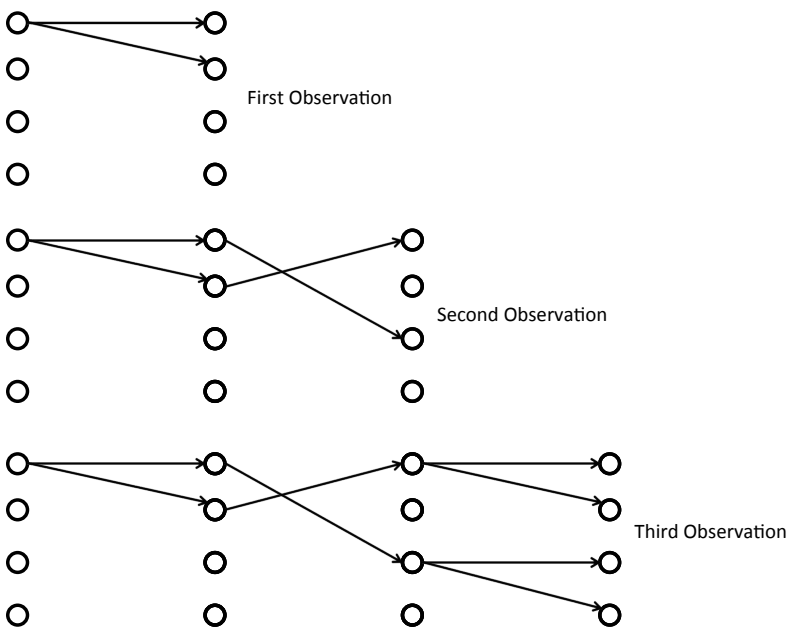


Recursive State Estimation
Output State Mappings and Induced Trellis Diagrams

- Each observable event $\sigma_o \in \Sigma_o$ can be associated with a state mapping

$$m_{\sigma_o} = \{(x_c, x_n) \mid \exists s \in \Sigma^* \text{ s.t. } P(s) = \sigma_o \text{ and } x_n \in \delta(x_c, s)\}$$

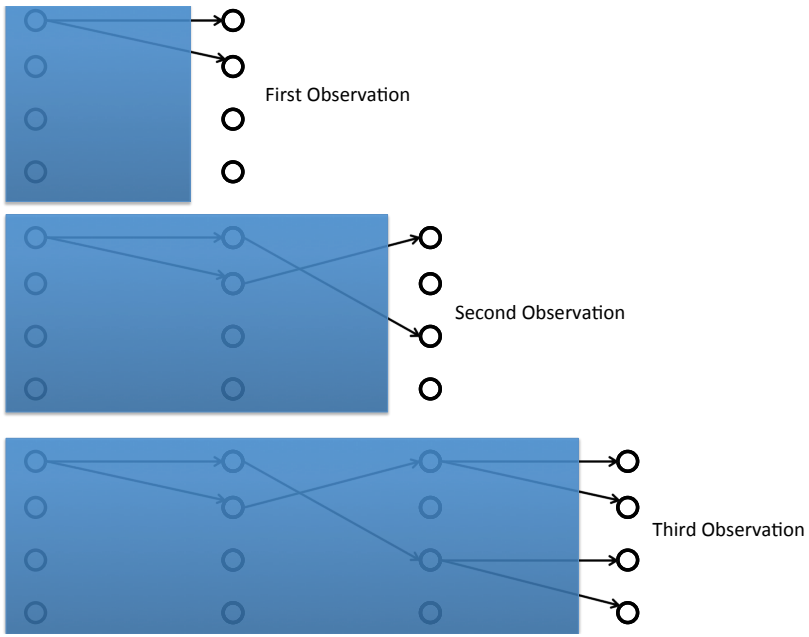
- State mappings corresponding to a sequence of observable events, $\omega = \sigma_{i_1}\sigma_{i_2}...\sigma_{i_k} \in \Sigma_o^*$ can be concatenated as $m_{\sigma_{i_1}} \bullet m_{\sigma_{i_2}} \bullet ... \bullet m_{\sigma_{i_k}}$
- Resulting construction captures matching sequences of states



- Arrows represent (possibly different) sequences of events that generate the observation at the corresponding stage

Current State Estimation
Prunning of Trellis Diagrams

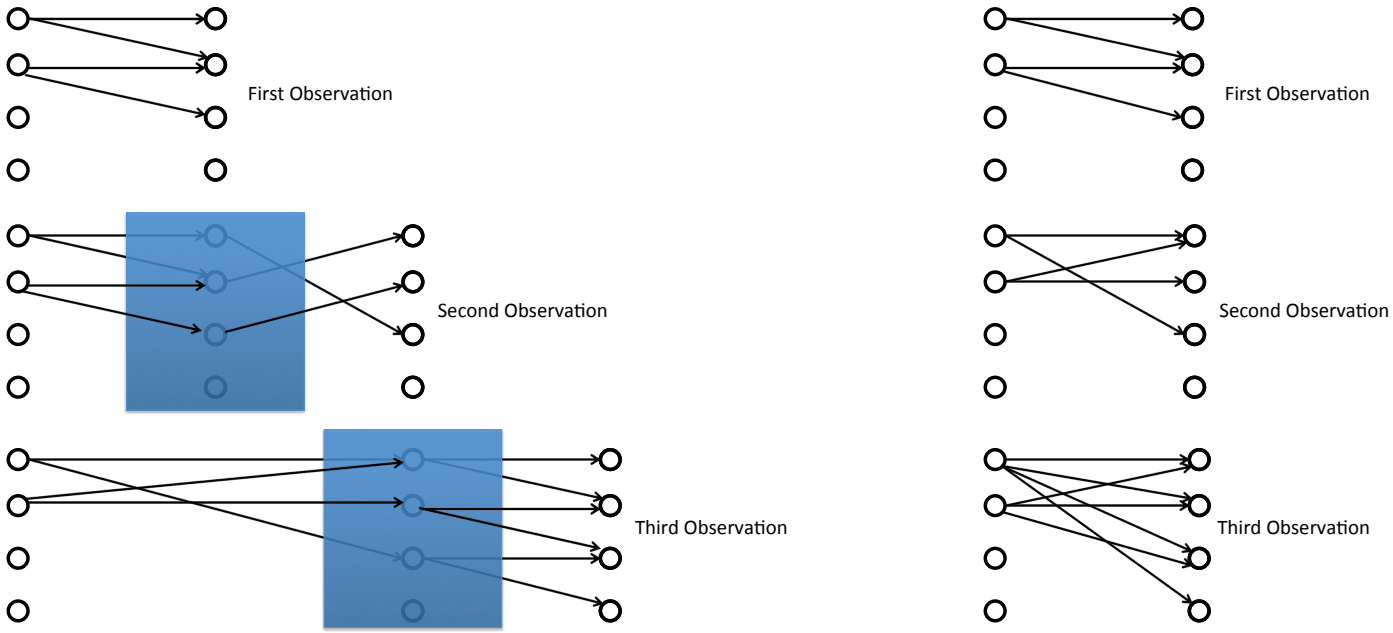
- Trellis diagrams can be pruned from parts not useful for task at hand
- Current state estimation only needs latest stage



- Results in earlier recursive current state estimation procedure
- One can also annotate possible current states with additional information (e.g., *a posteriori* probabilities for current states)

Initial State Estimation
Prunning of Trellis Diagram

- Initial state estimation only needs initial stage and latest stage



- Reduced construction (on the right) captures earlier recursive initial state estimation procedure
- Again, one can annotate possible initial/current states with additional information (e.g., *a posteriori* probabilities for initial states)

Talk Outline

- Processing of Observations (centralized)
  1. Recursive current state estimation
  2. Generalizations to initial/delayed state estimation
- Properties of Interest and their Verification
  1. Detectability
  2. Fault detection and diagnosis
  3. Current-state and initial-state opacity
- Decentralized Observation Settings (no communication between sites)
  - Fusion of information at coordinator
  - Conveying decisions, or estimates, or observation sequences
  - Verification of properties of interest
- Distributed Observation Settings (some communication between sites)

Strong Detectability
Eventual Knowledge of Exact System State

- Given: NFA $G = (X, \Sigma, \delta, X_0)$ with observable events Σ_o ($\Sigma_o \subseteq \Sigma$)

- Strong Detectability [Shu et al, 2007]: G is strongly detectable if for *all* event sequences $s \in L(G)$ that generate observations longer than a critical length N_c, one can determine exactly the current state of the system; formally,

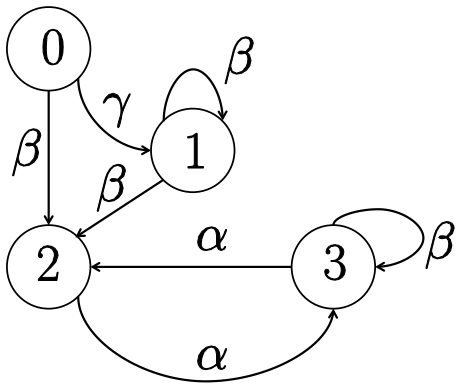
$$\exists N_c \in \mathbb{N}, \forall s \in L(G) \; \{ | \underbrace{P(s)}_{\omega} | \geq N_c \Rightarrow |\widehat{X}(\omega)| = 1 \}$$

- Conditions imposed on state estimates: Eventually, all state estimates become singleton subsets of X
  Variations: Strong/weak detectability; strong/weak periodic detectability

- Generalization: State isolation (captures whether system state can be isolated, eventually, within subsets with specific properties)

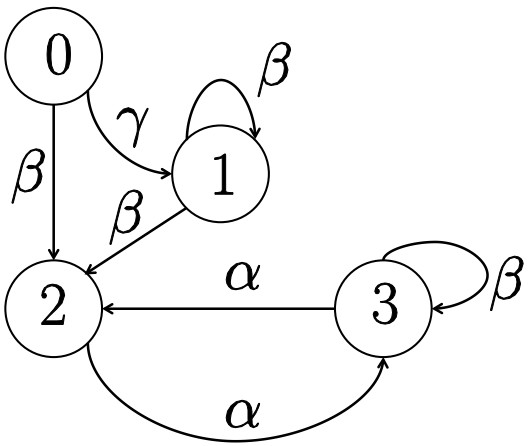
- Challenge: Verification has to consider what happens for all possible sequences of observations (as early as [Lin and Wonham, 1988], [Ozveren and Willsky, 1990], [Caines et al, 1991])

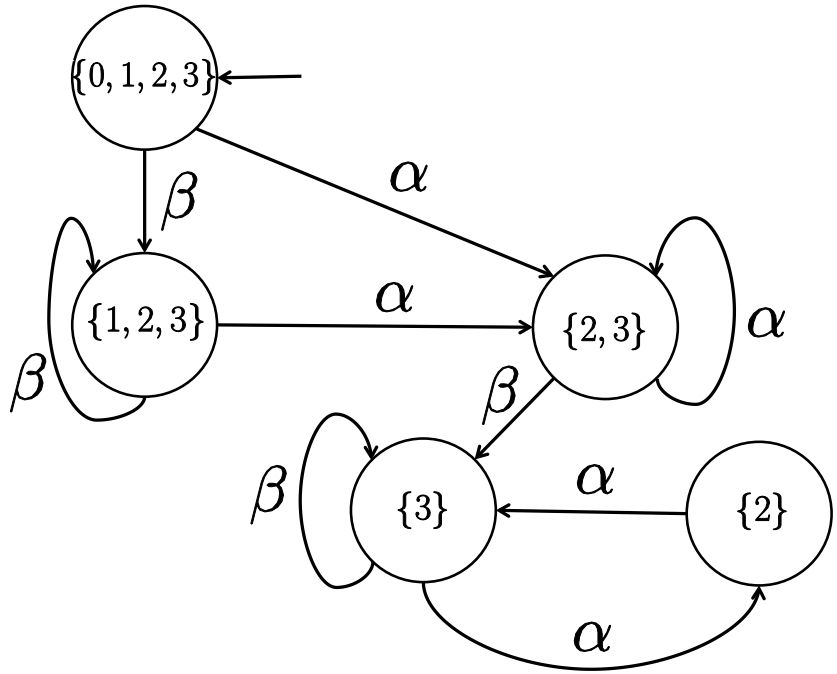
Strong Detectability
Example



- Consider NFA G used earlier with $X_0 = \{0, 1, 2, 3\}$ and $\Sigma_o = \{\alpha, \beta\}$
- Recall that, if we observe $\omega = \alpha\beta\alpha$, we can infer after each observation
$$\left\{ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \right\} \xrightarrow{\alpha} \left\{ \begin{array}{c} 2 \\ 3 \end{array} \right\} \xrightarrow{\beta} \{\ 3\ \} \xrightarrow{\alpha} \{\ 2\ \}$$
- At this point, we know the state exactly; in fact, we will know it exactly from this point on, for all continuations: $(\alpha\beta^*\alpha)^*$, or $(\alpha\beta^*\alpha)^*\alpha$, or $(\alpha\beta^*\alpha)^*\alpha\beta^*$
- To verify strong detectability, we need to verify that something similar holds for all possible activity in the given system

Verification of Strong Detectability
Example of Observer Construction





- Need to check for cycles of observer states associated with more than one system state estimate
  E.g., cycle $\beta\beta^*$ at state $\{1, 2, 3\}$ implies the system is not detectable
- Absence of such cycles $\Leftrightarrow$ Strong detectability [Shu et al, 2007]
- Absence of cycles can be checked with complexity that is polynomial in the size of the observer; however, observer itself can have *exponential* complexity in $|X|$

Formal Construction of Observer

Determinizing an NFA [Cassandras and Lafortune, 2008]

- Given: NFA $G = (X, \Sigma, \delta, X_0)$ with observable events Σ_o ($\Sigma_o \subseteq \Sigma$)
- Observer (or Current-State Estimator): *Deterministic* finite automaton $G_{obs} = (Q_{obs}, \Sigma_o, \delta_{obs}, Q_{0,obs})$ constructed as follows:
  1. $Q_{obs} \subseteq 2^X$, i.e., each observer state $q_{obs} \in Q_{obs}$ is associated with a unique subset of states of the given NFA G, i.e., $q_{obs} \subseteq X$
  2. Initial state is $Q_{0,obs} = R(X_0, \epsilon)$ (unobservable reach of X_0)
  3. From any state $q_{obs} \in Q_{obs}$ (recall $q_{obs} \subseteq X$) of the current-state estimator, the next state for any $\sigma_o \in \Sigma_o$ is captured by

$$\delta_{obs}(q_{obs}, \sigma_o) = R(q_{obs}, \sigma_o)$$

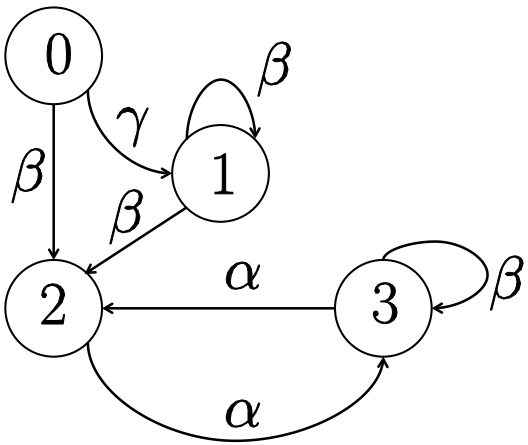
- Observer captures the set of possible current states in G following a sequence of observations $\omega \in \Sigma_o^*$ via

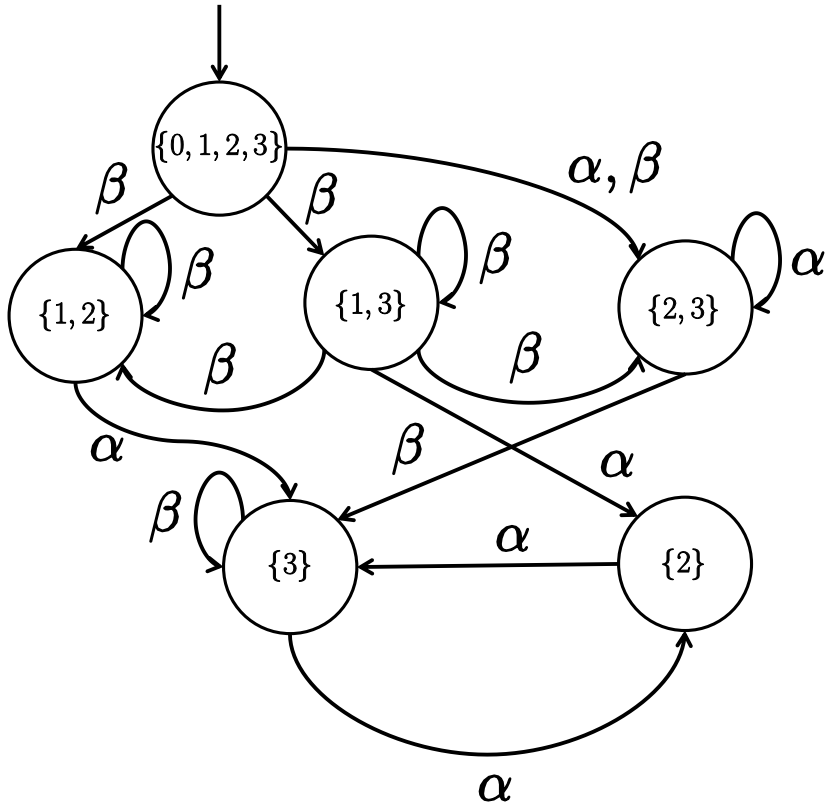
$$\boxed{\widehat{X}(\omega) = \delta_{obs}(Q_{0,obs}, \omega)}$$

- Note: Observer not needed for online state estimation, but can be convenient for verification of certain properties (in some cases, verification may be possible via less complex constructions)

Efficient Verification of Strong Detectability

Example of Detector Construction (Tracking Pairwise Uncertainty [Shu and Lin, 2010])





- Need to check for cycles of detector states that are associated with sets of more than one system state estimate
  E.g., cycle $\beta\beta^*$ at state $\{1,2\}$ implies the system is not detectable
- Absence of such cycles $\Leftrightarrow$ Strong detectability [Shu and Lin, 2010]
- Absence of cycles can be checked with complexity that is polynomial in the size of the detector; complexity of detector *polynomial* in $|X|$

Formal Construction of Detector

Tracking Pairwise Uncertainty Sets [Shu et al, 2007]

- Given: NFA $G = (X, \Sigma, \delta, X_0)$ with observable events Σ_o ($\Sigma_o \subseteq \Sigma$)
- Detector: Nondeterministic finite automaton
  $G_{det} = (Q_{det}, \Sigma_o, \delta_{det}, Q_{0,det})$ constructed as follows:
  1. $Q_{det} = Q_p \cup Q_s \cup Q_{0,det}$ is the finite set of states, with
     1) $Q_p = \{\{x_l, x_m\} \mid x_l, x_m \in X, x_l \neq x_m\}$ (pairs of different states)
     2) $Q_s = \{\{x_j\} \mid x_j \in X\}$ (singleton sets of states)
     3) $Q_{0,det} = R(X_0, \epsilon)$ (unobservable reach of X_0)
  2. From any state $q_d \in Q_{det}$ of the detector, the next state(s) for any $\sigma_o \in \Sigma_o$ is captured by $\delta_{det} : Q_{det} \times \Sigma_o \to 2^{Q_{det}}$ as

$$\delta_{det}(q_d, \sigma_o) = \begin{cases} \{q_p \in Q_p \mid q_p \subseteq R(q_d, \sigma_o)\}, & \text{if } |R(q_d, \sigma_o)| > 1 \\ \{x_l\} \in Q_s, & \text{if } R(q_d, \sigma_o) = \{x_l\} \\ \text{undefined}, & \text{if } R(q_d, \sigma_o) = \emptyset \end{cases}$$

- Detector requires at most $|X|^2 + 1$ states ($\Rightarrow$ strong detectability can be checked with polynomial complexity)
- Unlike the observer, the detector is a nondeterministic automaton

Fault Detection
Eventual Detection of Event Occurrence

- Fault events Σ_f ($\Sigma_f \subseteq \Sigma_u$ where Σ_u are the unobservable events)
- Fault Detection: Determine, perhaps after some bounded delay, the occurrence of a *fault* event $\sigma_f \in \Sigma_f$
- Can be viewed as a special case of an event inference task
- Diagnosability (single class) [Sampath et al, 1995]: System allows external observer to determine, with some *bounded* delay, the occurrence of any *fault* event; formally,
  $(\exists N_c \in \mathbb{N})(\forall (s\sigma_f) \in L(G), s \in (\Sigma \setminus \Sigma_f)^*, \sigma_f \in \Sigma_f)(\forall t \in L(G)/(s\sigma_f))$, s.t.

  $$[\ |t| \geq N_c \Rightarrow D(s\sigma_f t)\ ]\ ,$$

  where the diagnosability function $D : \Sigma^* \to \{0, 1\}$ is

  $$D(s\sigma_f t) = \begin{cases} 1, & \text{if } s' \in P^{-1}[P(s\sigma_f t)] \Rightarrow \sigma_f \in s' \\ 0, & \text{otherwise} \end{cases}$$

  i.e., all "explanations" include σ_f ($\Rightarrow \sigma_f$ has occurred with certainty)
- Delay needed to detect fault might be different for different faults

Generalizations to Fault Diagnosis

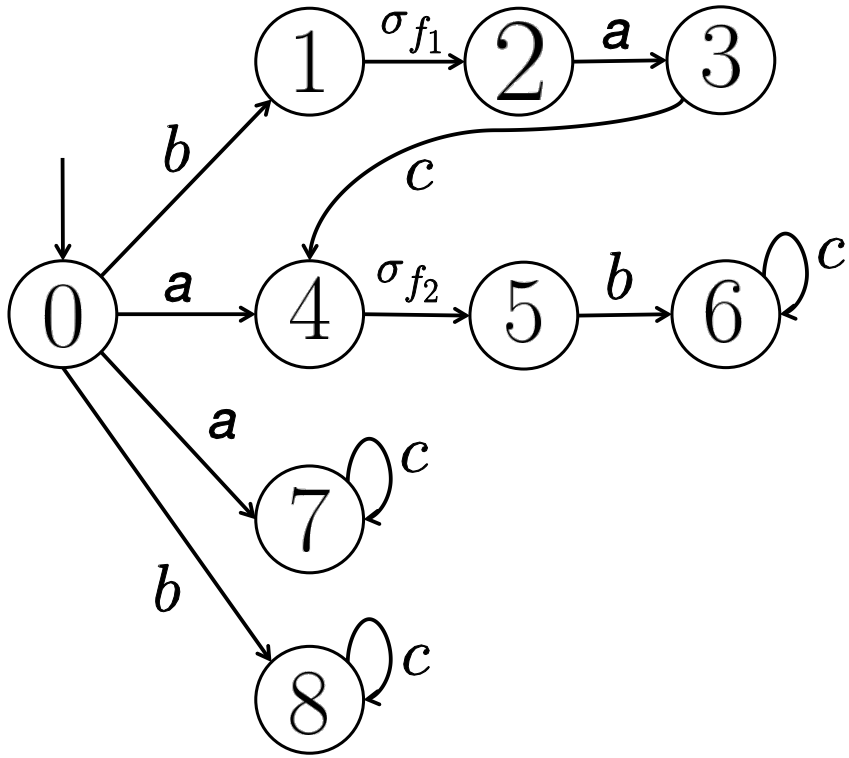
Eventual Classification of Event Occurence [Sampath et al, 1995]

- $\Sigma_f = \Sigma_{f_1} \dot{\cup} \Sigma_{f_2} \dot{\cup} \ldots \dot{\cup} \Sigma_{f_K}$, where K is the number of different fault classes
- Fault Diagnosis: Ability to determine, perhaps after some bounded delay, the occurrence of a *fault* event from a particular class
- Numerous variations (e.g., tracking faults from multiple fault classes or order of faults from different classes, some discussions later)
- Extensive activity for two decades, but several open problems remain:
  1. Distributed/hierarchical settings, information exchange strategies
     E.g., work by Benveniste, Fabre, Kumar, Su, and others
     Some discussions later
  2. Sensor selection (static, off-line) or activation (dynamic, on-line)
     E.g., work by Biswas, Lafortune, Rudie, Teneketzis, and others
     Not discussed

- $\Sigma_f = \Sigma_u = \{\sigma_{f_1}, \sigma_{f_2}\}$
- $X_0 = \{0\}$
- If sequence of events $a\sigma_{f_2}b$ occurs, we observe ab and we infer

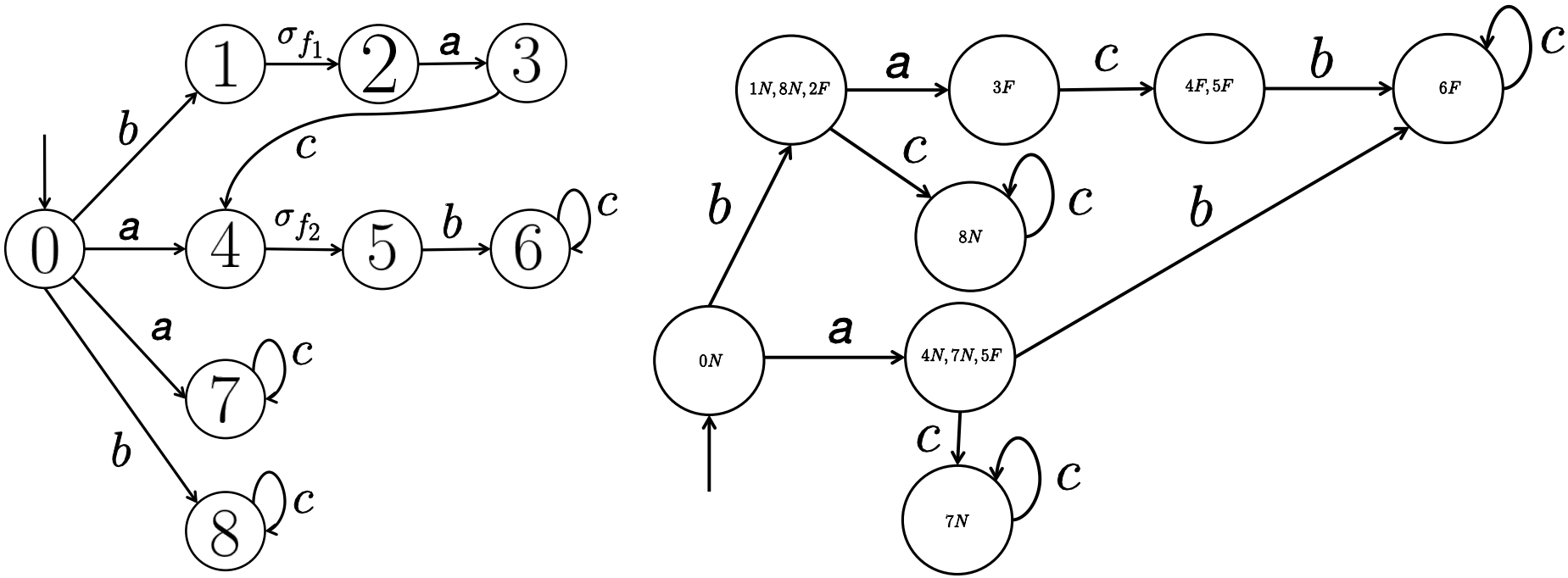
$$\{\ 0N\ \} \xrightarrow{a} \left\{ \begin{array}{l} 4N \\ 5F \\ 7N \end{array} \right\} \xrightarrow{b} \{\ 6F\ \}$$

Fault σ_{f_2} detected after one event!



- Labels N ("normal") and F ("fault") annotate state estimates to capture occurrence of fault events (sometimes we drop label N)
- Notion of diagnosability (i.e., systematic detection of all faults) can be checked via a diagnoser or a verifier

Fault Detection
Example of Diagnoser Construction

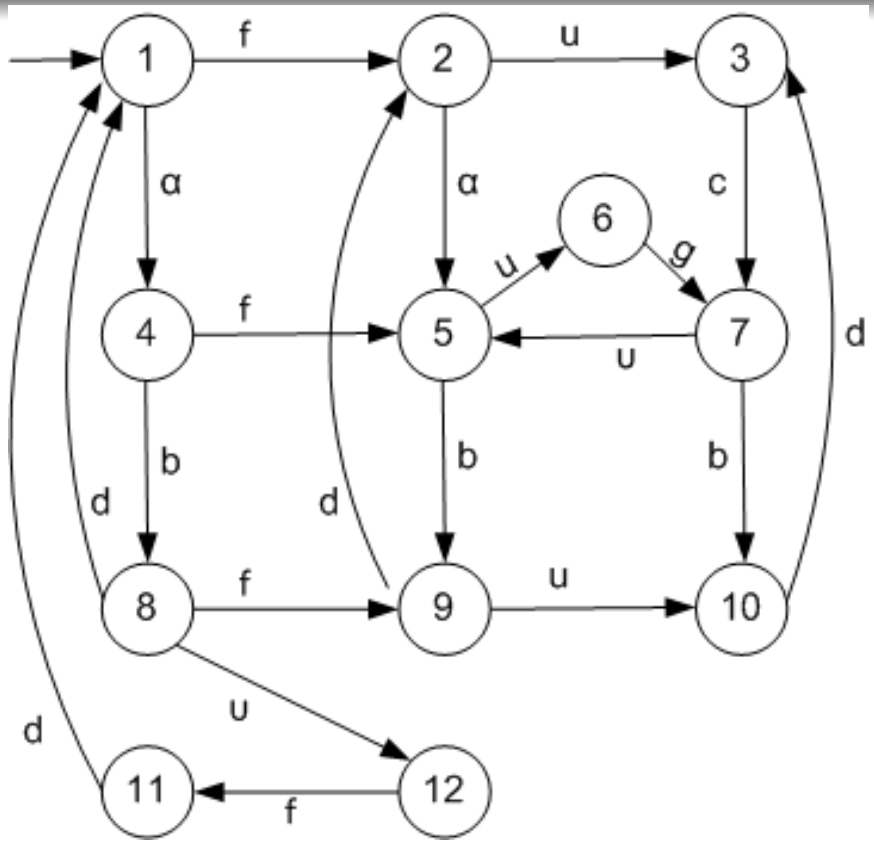


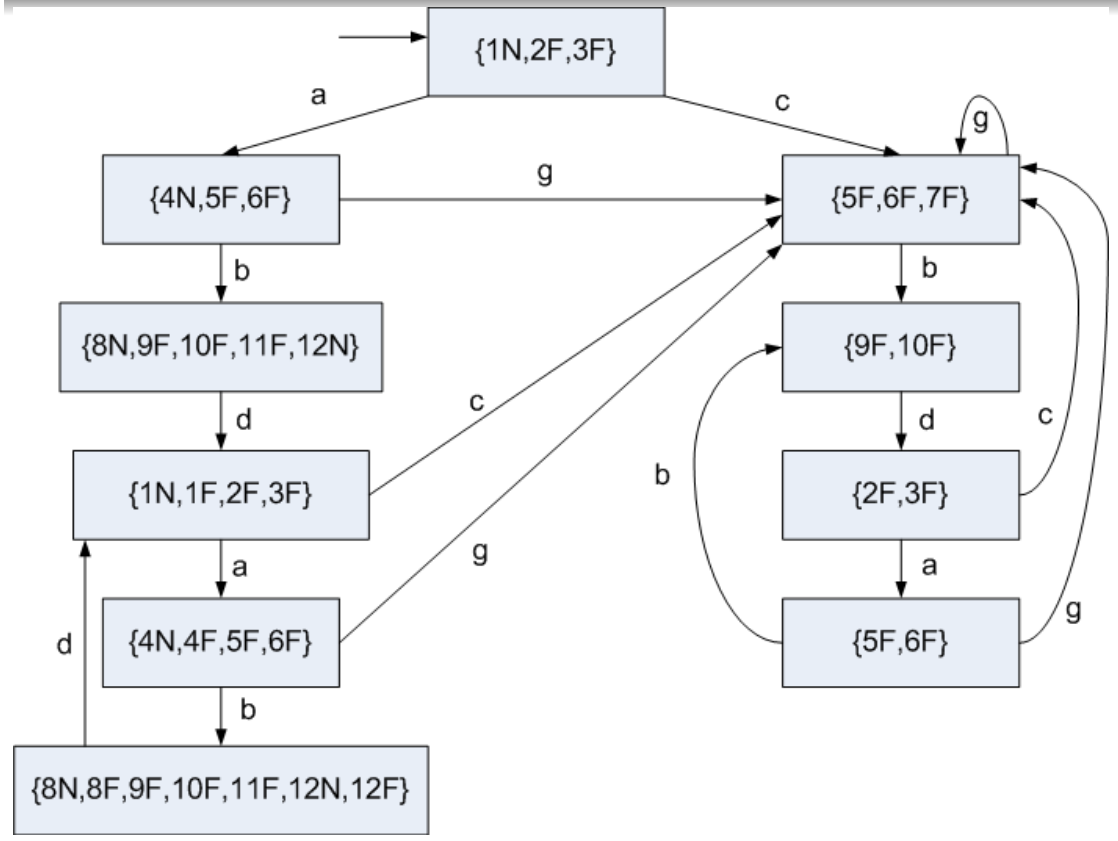
Unlike detectability:

- Only concerned about what happens after a fault
- No need to eventually reach singleton sets of state estimates
- Need absence of indeterminate cycles, i.e., cycles that involve *uncertain* diagnoser states (whose state estimates involve both *N* and *F* labels)

Fault Detection

Larger Example of Diagnoser Construction (from [Cassandras and Lafortune, 2008])





- $\Sigma_f = \{f\}$, $\Sigma_u = \{f, u, \upsilon\}$, $X_0 = \{1\}$
- Observation sequences of the form $(abd)^*$ do not allow us to determine that a fault has occurred (cannot guarantee detection with finite delay)
- Cycles through uncertain states may not necessarily be problematic (need to correspond to actual cycles of events that can occur after a fault event in the system)

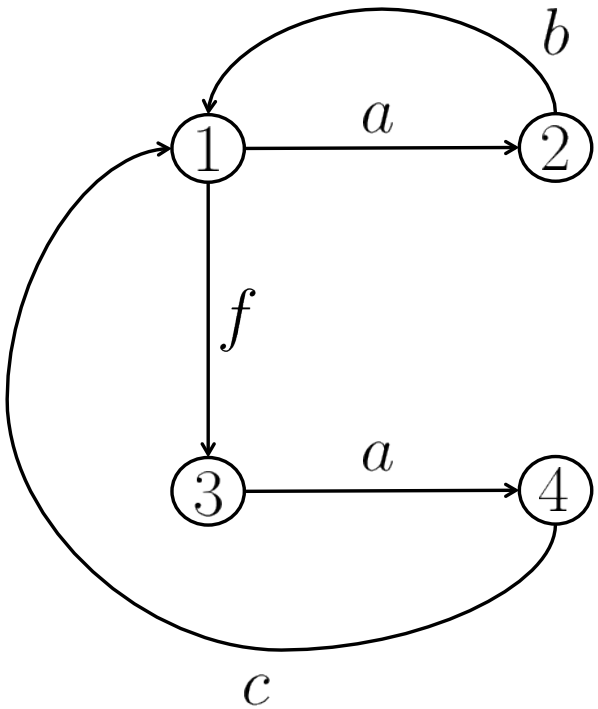
Indeterminate Cycles Not Always Problematic

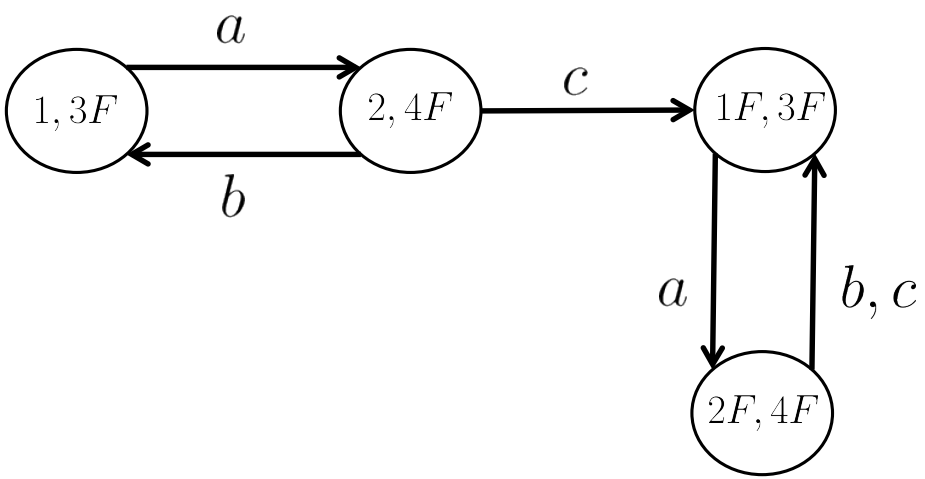
Example of Non-Problematic Indeterminate Cycle [Hadjicostis, 2018]

Consider NFA G below (left) where $\Sigma_f = \Sigma_u = \{f\}$ and $X_0 = \{1\}$, and its diagnoser (right)

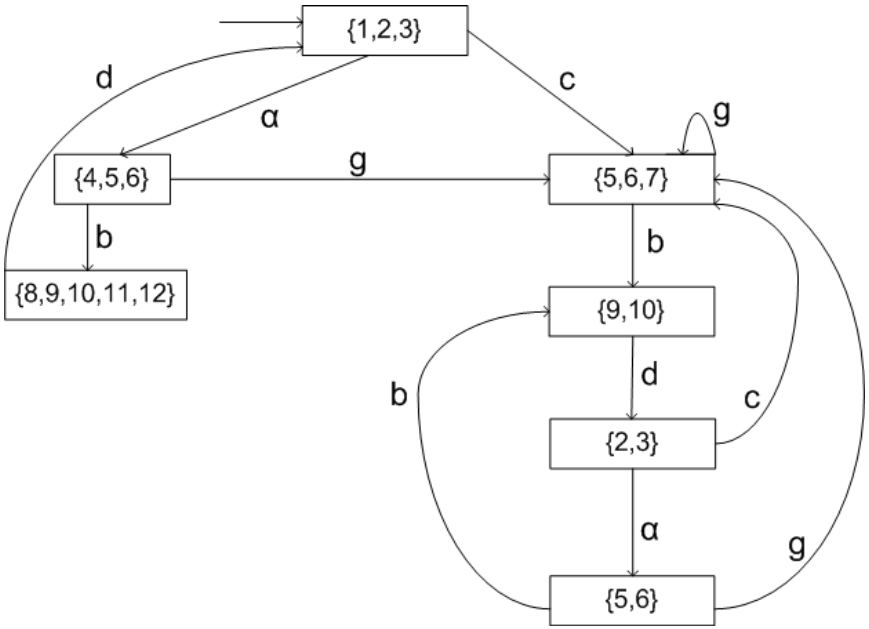
Indeterminate cycle in diagnoser is *not* problematic (because it is *not* executable after the fault)

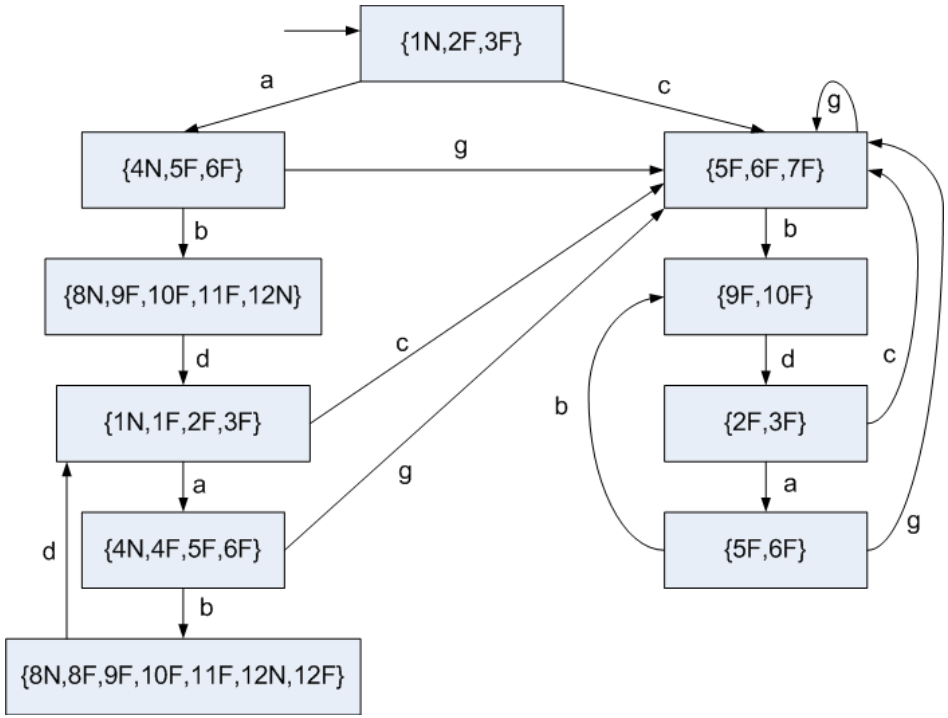
$\Rightarrow$ Need to check that indeterminate cycle is executable after the fault





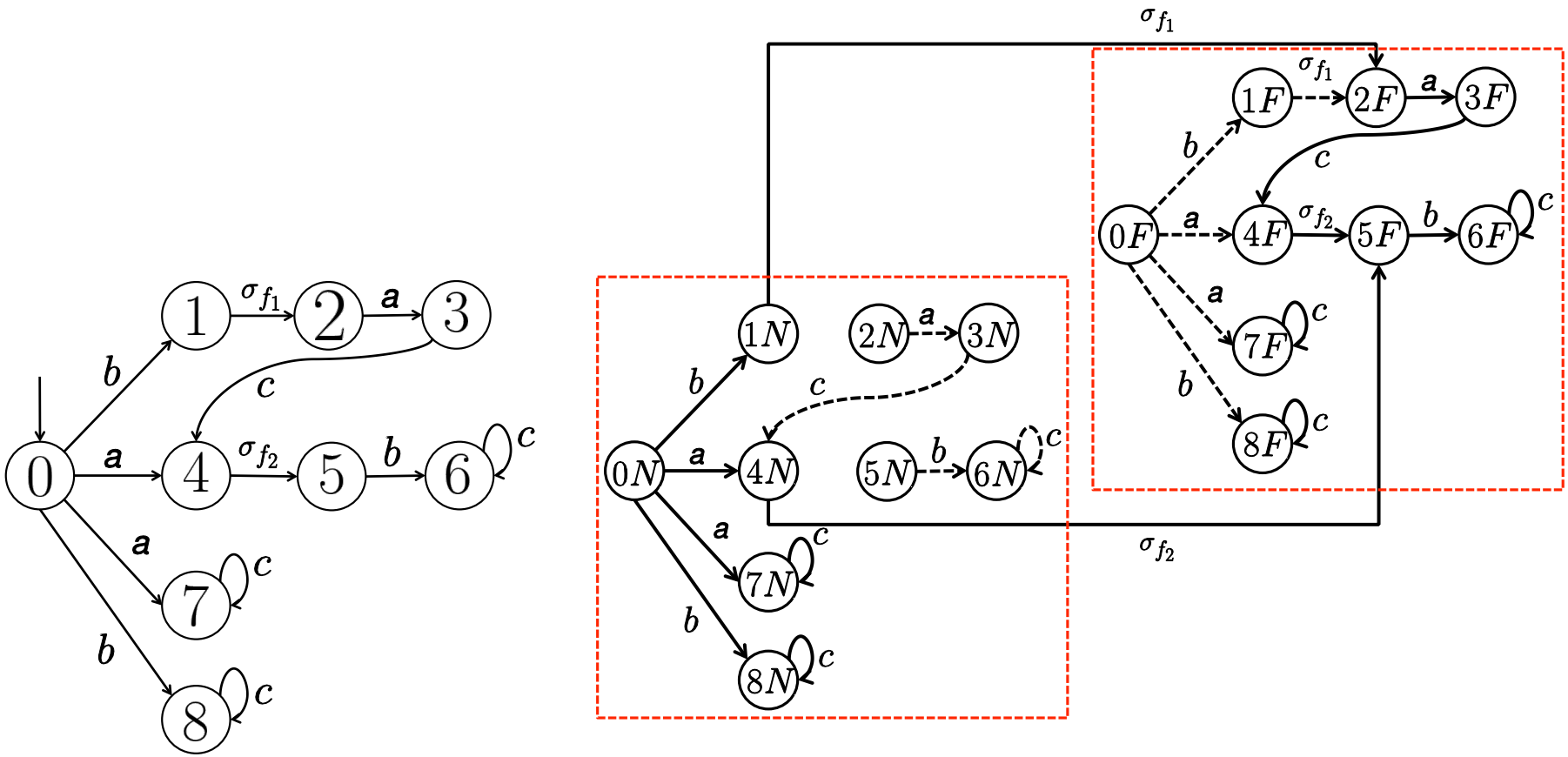
Observer vs Diagnoser
Adding Labels to States





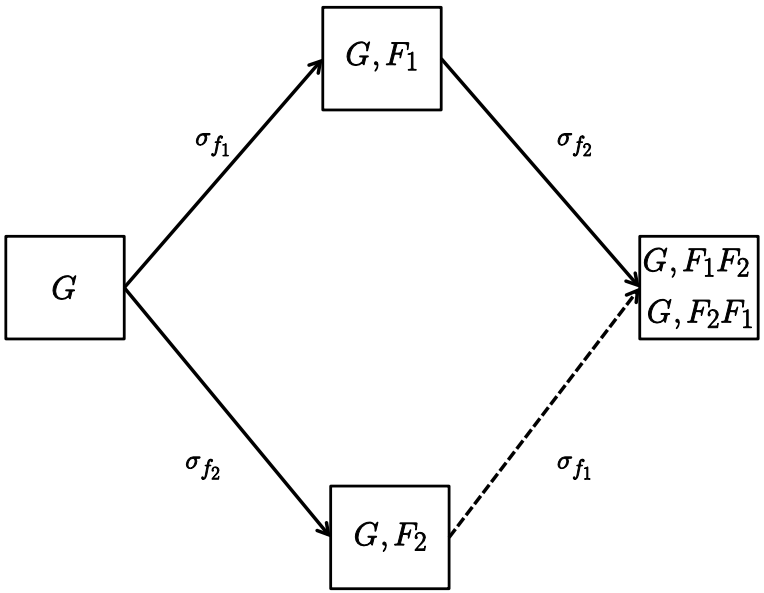
- The diagnoser states are refined versions of the observer states
- Observer has potentially $2^{|X|}$ states whereas diagnoser has potentially $2^{2|X|} = 4^{|X|}$ states

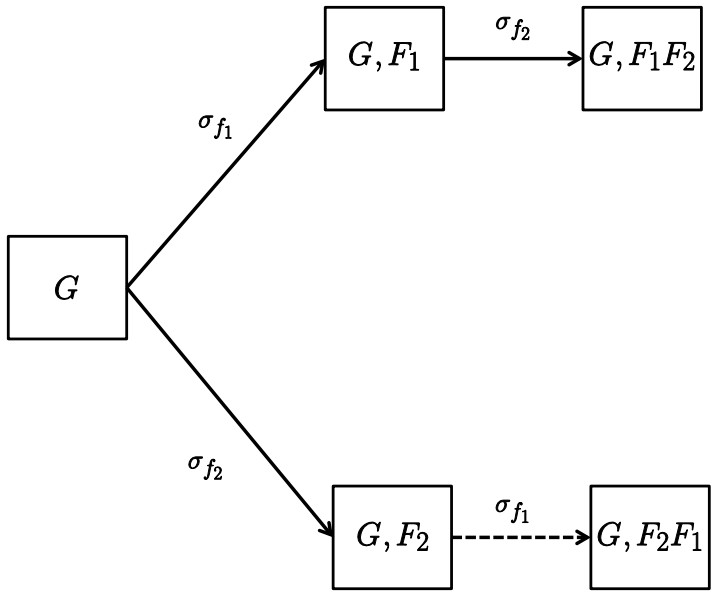
Fault Detection
Connections to State Estimation



- Diagnoser: Observer for the modified G_F on the right
- Pairs of states/labels are referred to as “diagnostic information"

Fault Diagnosis
Connections to State Estimation via Different Types of G_F





- Left: Modified G_F for tracking occurrence of faults σ_{f_1} and σ_{f_2} (whether σ_{f_1} or σ_{f_2}, or both, have occurred)
- Right: Modified G_F for tracking occurrence of faults σ_{f_1} and σ_{f_2} (whether σ_{f_1} or σ_{f_2}, or both, have occurred, as well as the *order* in which they have occurred)

NFA $G = (X, \Sigma, \delta, X_0)$ with observable events Σ_o and faults $\Sigma_f \subseteq \Sigma_u$

Using Diagnoser (DFA with exponential number of states)

- Diagnoser can be seen as observer on G_F
  1. States are subsets of $X \times L$
  2. Label set L (e.g., $L = \{N, F\}$ where N=normal and F=faulty)
- Uncertain state: A diagnoser state $q_d \subseteq 2^{X \times L}$ for which $\exists (x, l), (y, l') \in q_d$, such that $F \in l$ and $F \notin l'$
- Diagnosability relates to existence of indeterminate cycles (cycles of uncertain states) in diagnoser [Sampath et al, 1995]

Using Verifier (NFA with polynomial number of states)

- Verifier relates to detector of G_F with states that are pairs of states of the form $(x_i, l_i), (x_j, l_j)$, $x_i, x_j \in X$, $l_i, l_j \in L$
- Uncertain state: A verifier state $(x_i, l_i), (x_j, l_j)$ for which $F \in l_i$ and $F \notin l_j'$
- Diagnosability relates to existence of indeterminate cycles (cycles of uncertain states) in verifier [Jiang et al, 2001], [Yoo and Lafortune, 2002]

Current State Opacity

Definition and Problem Description [Saboori and CNH, 2007, 2011]

- Given: NFA $G = (X, \Sigma, \delta, X_0)$ with observable events Σ_o ($\Sigma_o \subseteq \Sigma$) and subset of secret states S ($S \subseteq X$)

- Current state opacity requires that an external observer can never be certain that system state is within the set of secret states S
  [At least one state outside S is possible; relates to "possible innocence" in anonymity protocols]

- Current State Opacity [Saboori and CNH, 2007, 2011]: For all $s \in L(G)$, for all $x_0 \in X_0$, it holds
  $\{\delta(x_0, s) \subseteq S\} \Rightarrow \{\exists t \in \Sigma^*, \exists x_0' \in X_0, \{P(t) = P(s), \delta(x_0', t) \notin S\}\}$

- Verification using an observer: Check that no observer state q_{obs} satisfies $q_{obs} \subseteq S$

- Extensive activity in the last ten years (e.g., language-based opacity, enforcement, probabilistic settings, game-theoretic formulations, etc., by Darondeau, Dubreil, Faure, Kumar, Lafortune, Lesage, Lin, Saboori, Marchand, Takai, and others)

Initial State Opacity

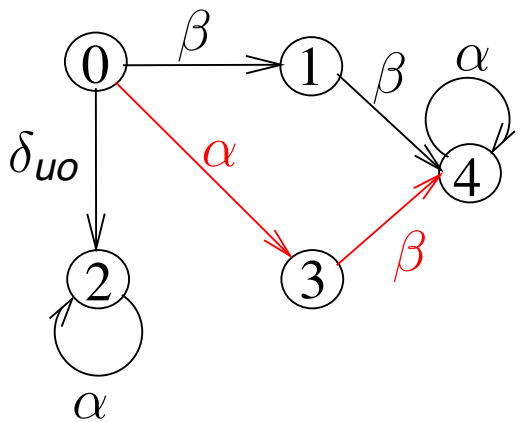
Definition and Problem Description [Saboori and CNH, 2008]

- Given: NFA $G = (X, \Sigma, \delta, X_0)$ with observable events Σ_o ($\Sigma_o \subseteq \Sigma$) and subset of secret states S_0 ($S_0 \subseteq X_0$)
- Initial state opacity requires that an external observer can never be certain that system initial state is within the set of secret initial states S_0 [For all observation sequences, at least one state outside S_0 is possible]
- Initial State Opacity [Saboori and CNH, 2008]: For all $s \in L(G)$, for all $x_0 \in S_0$, it holds
  $\{\delta(x_0, s) \neq \emptyset\} \Rightarrow \{\exists t \in \Sigma^*, \exists x_0' \in (X_0 \setminus S_0), \{P(t) = P(s), \delta(x_0', t) \neq \emptyset\}\}$
- Verification using an initial state estimator

Initial State Opacity
Example of Violation

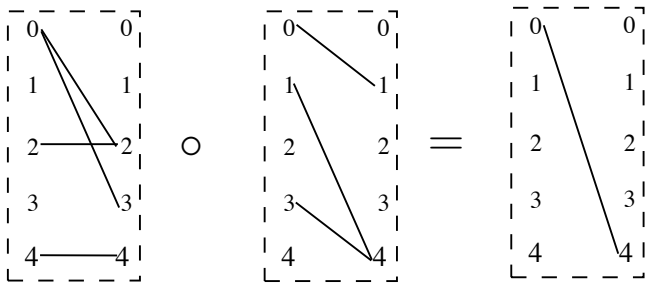
System specifications:

- $X_0 = X = \{0, 1, 2, 3, 4\}$
- $\Sigma = \{\alpha, \beta, \delta_{uo}\}$
- $\Sigma_u = \{\delta_{uo}\}$

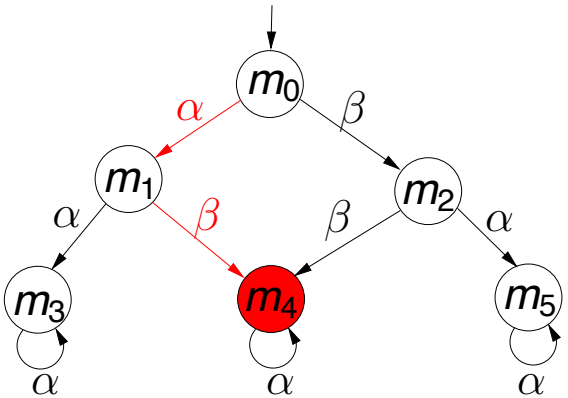


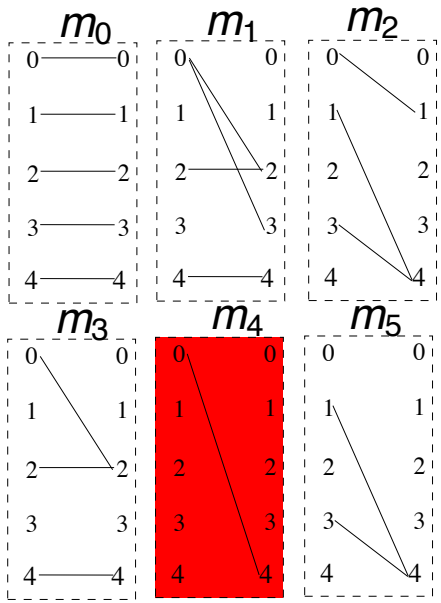
Secret initial states $S_0 = \{0\}$

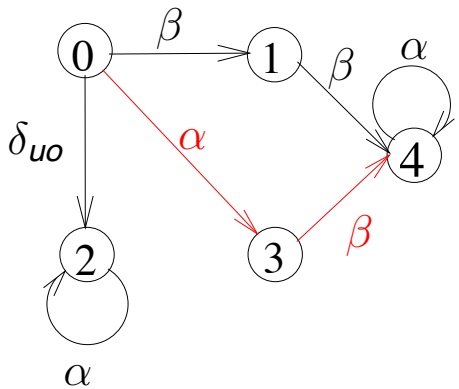
- Violation after observing $\alpha\beta$



Initial State Opacity
Verification Using Initial State Estimator







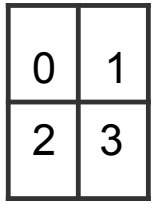
- $X_0 = X = \{0, 1, 2, 3, 4\}$, $\Sigma = \{\alpha, \beta, \delta_{uo}\}$, $\Sigma_u = \{\delta_{uo}\}$
- With $S_0 = \{0\}$, we have a violation of initial state opacity
  Observation sequences of the form $\alpha\beta\alpha^*$ (or $\beta\beta\alpha^*$) lead to a violation
- With $S_0 = \{2\}$, we do not have a violation of initial state opacity
  Each time state 2 appears as a possible initial state, other initial states are also possible

Motivational Example
Coverage Limitations in Sensor Networks

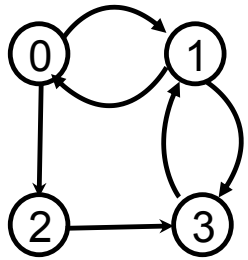
- Two-dimensional grid with vehicle state corresponding to cell number of its location
- Kinematic Model: Describes limitations on vehicle movements due to physical obstacles on the grid or due to logical constraints
- Enhanced Kinematic Model: Assigns label ℓ to all transitions that end in a cell covered by sensor ℓ
- Certain locations (in secret set S) in the grid are sensitive and presence of vehicle in these locations should not be exposed
- Analysis/enforcement of such security/privacy considerations

Vehicle in a Toy 2D Grid

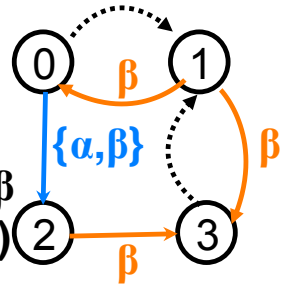
2-D Grid



Kinematic Model



Enhanced Kinematic Model (sensor α covers state 2 and sensor β covers states 0,2,3)



- Can the current cell (position) of the vehicle be identified via observations from the sensor network (for all or some trajectories)?
- How many observations are needed to identify the current cell (position)?
- Can the current location of the vehicle be pinpointed within a certain subset of secret locations?
- Similar questions can be asked about the initial location or past location (initial-state opacity, K-step opacity, etc.)
- To enforce opacity:
  - Supervisory control to disable behavior that exposes the user [Badouel et al, 2006]
  - Insertion labels may be used at the sensor to misguide the intruder/observer [Wu and Lafortune, 2014]
- Probabilistic knowledge, if available, can also be incorporated

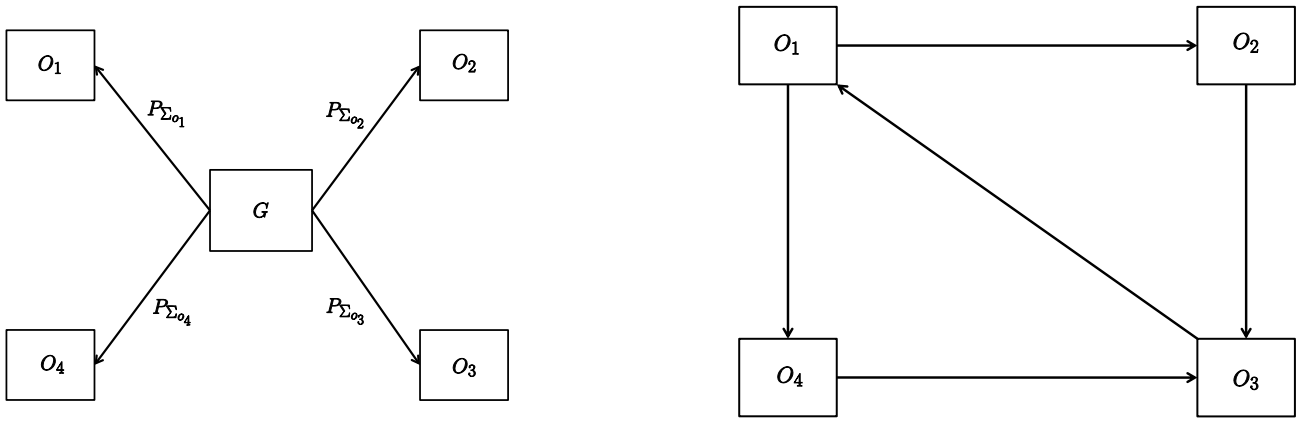
- Strong (periodic) detectability is easy (polynomial)
- Diagnosability is easy (polynomial)
- Current state opacity and initial state opacity are hard (NP-complete)
- Probabilistic current state opacity (can be defined in various ways) can be even undecidable

Talk Outline

- Processing of Observations (centralized)
  1. Recursive current state estimation
  2. Generalizations to initial/delayed state estimation
- Properties of Interest and their Verification
  1. Detectability
  2. Fault detection and diagnosis
  3. Current-state and initial-state opacity
- Decentralized Observation Settings (no communication between sites)
  - Fusion of information at coordinator
  - Conveying decisions, or estimates, or observation sequences
  - Verification of properties of interest
- Distributed Observation Settings (some communication between sites)

Distributed State Estimation

Observation Structure and Communication Constraints [Debouk et al, 2000]



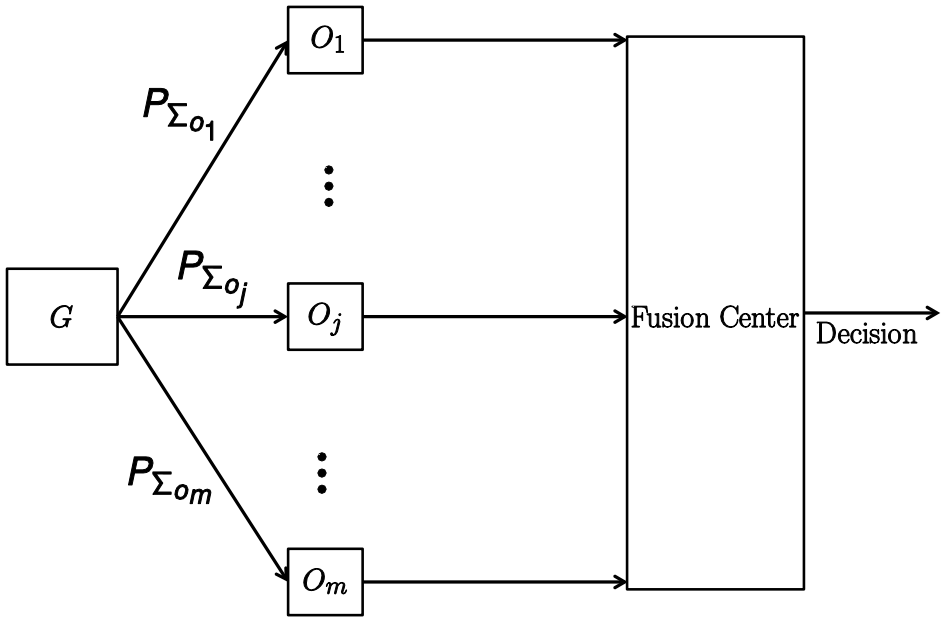
Example of distributed observation structure: NFA G observed via $m = 4$ observation sites (left) and communication topology between $m = 4$ observation sites (right)

- Multiple Observation Sites: O_1, O_2, ..., O_m, each with own set of observable events (observation site O_j observes events in Σ_{o_j}, $\Sigma_{o_j} \subseteq \Sigma$)
- Communication Topology: Communication constraints captured by directed graph $C = (V, E)$, with vertices $V = \{O_1, O_2, ..., O_m\}$, and edges $E \subseteq V \times V - \{(O_j, O_j) \mid O_j \in V\}$
  Edge (O_j, O_i) indicates that O_i can send information to O_j (directed link)
- Information Exchange: When/what information should be sent?

Note: Distributed observation structure could be the result of dealing with a modular system; we treat G as a monolithic system for simplicity

Fault Diagnosis in Decentralized Settings

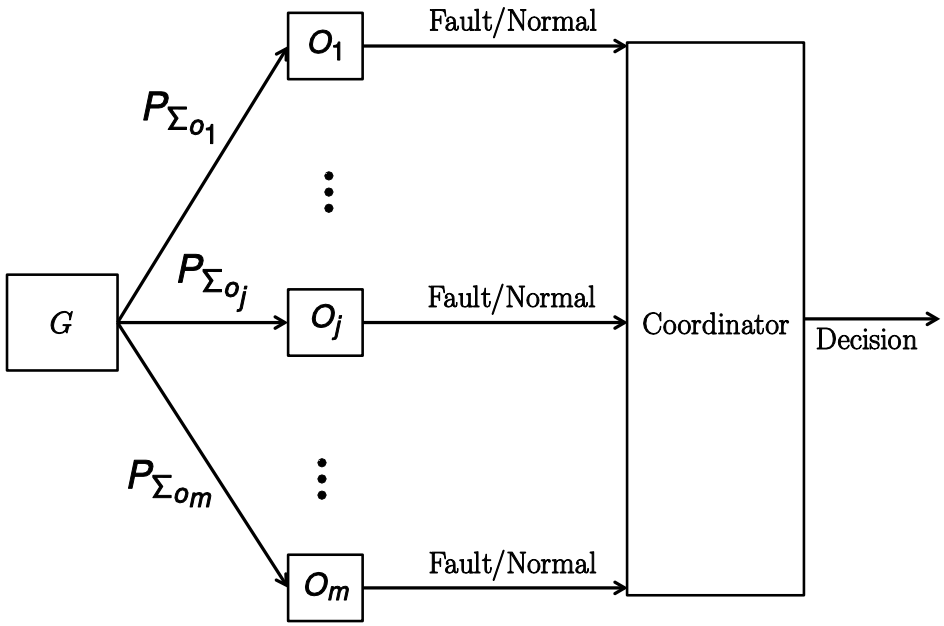
Special Case: Open Loop



- Open Loop: No feedback from fusion center to observation sites
- Event sequence $s \in L(G)$ induces $\omega_j = P_{\Sigma_{o_j}}(s)$ at site O_j
- Many parameters to be decided:
  1. Information to be communicated from observation sites
  2. Instants at which information is transmitted (simultaneously or not)
  3. Processing capability at observation sites and/or fusion center
  4. Synchronization issues (time stamps), delays, packet drops

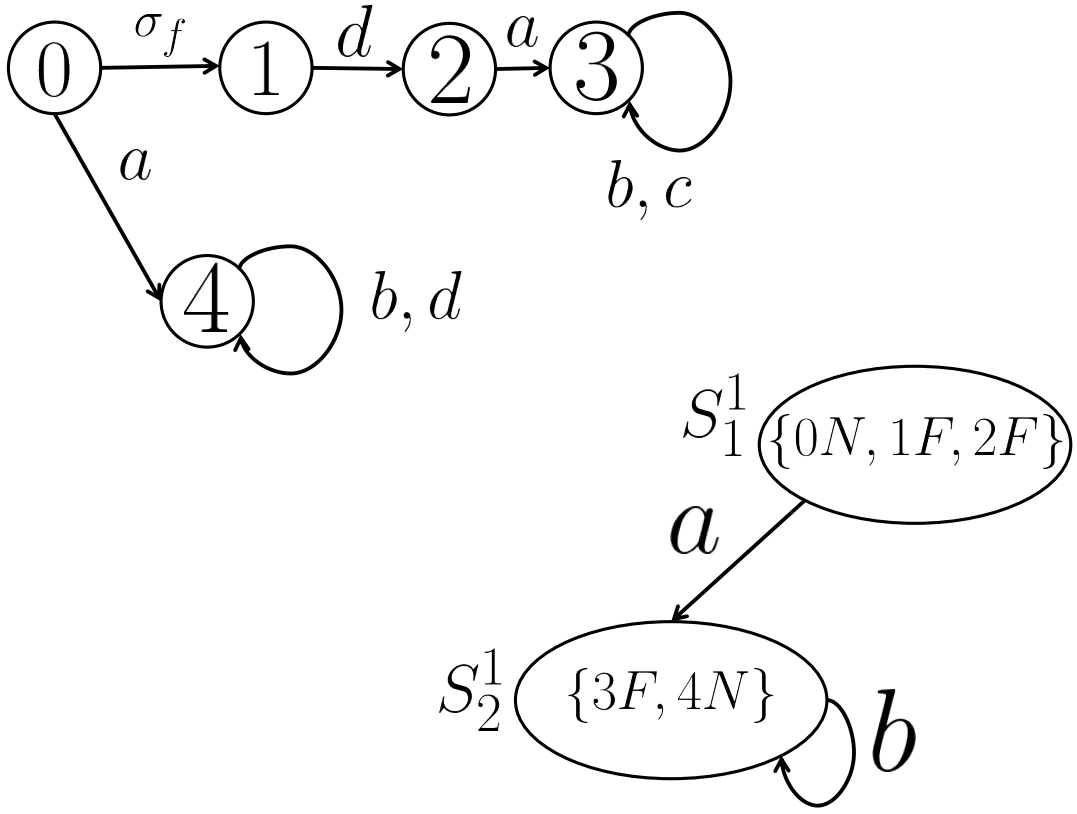
Fault Diagnosis in Decentralized Settings

Case 1: Codiagnosability



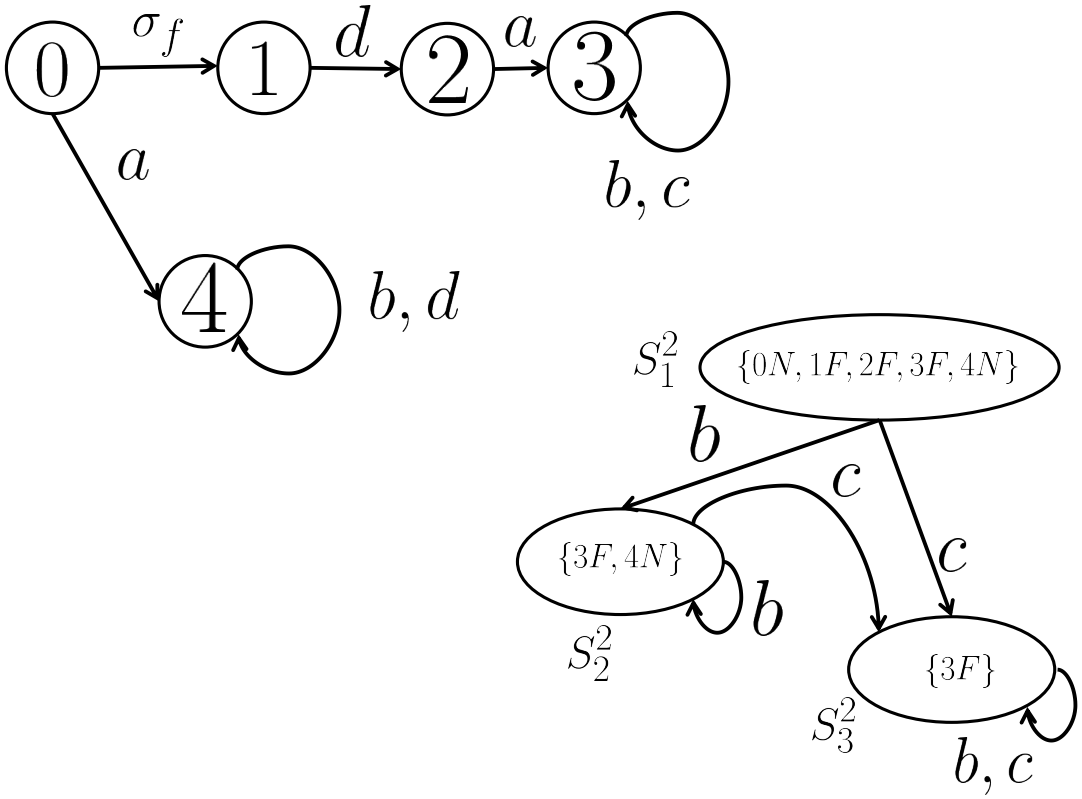
- Codiagnosability [Wang et al, 2007], [Qiu and Kumar, 2006]:
  1. Observations at O_j processed locally (based on Σ_{o_j})
  2. Coordinator declares a fault if at least one observation site diagnoses fault within bounded delay
  3. Variations of this basic setting also exist
- Problematic cases:
  Indeterminate cycles (of uncertain states) capture system behavior in which *all* local diagnosers remain confused

Example of Codiagnosability
Local Diagnosers (1)



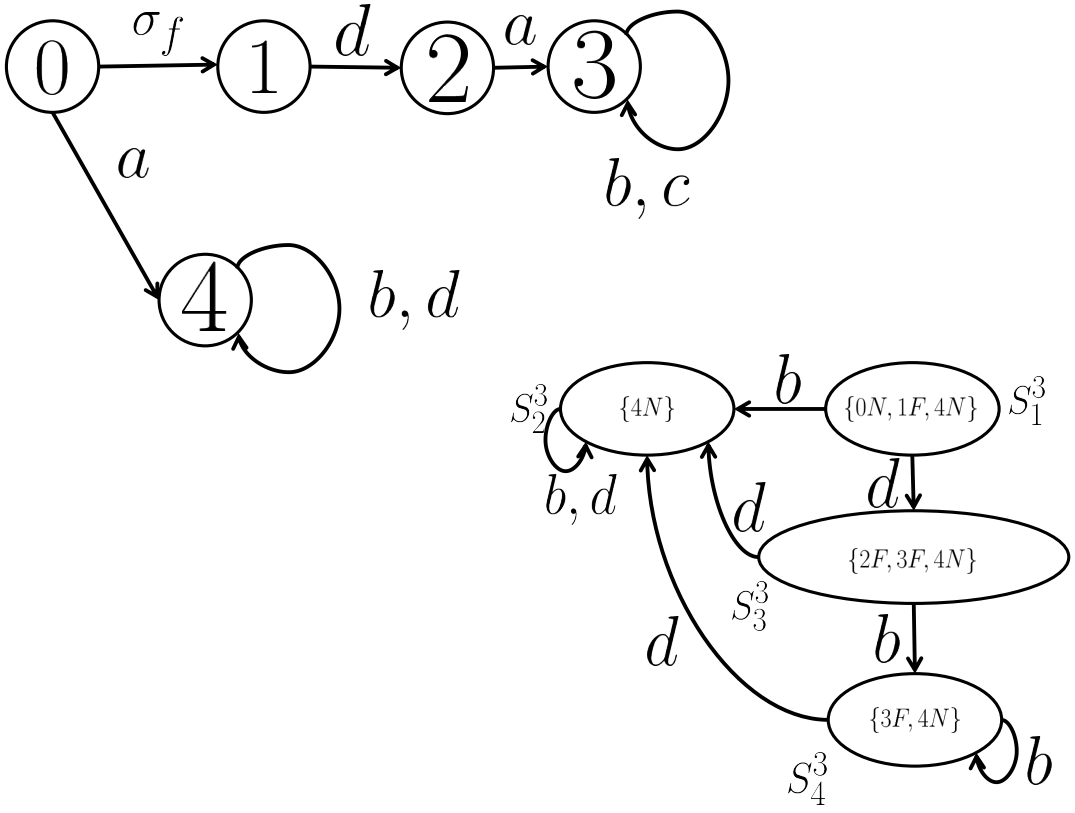
Local diagnoser D_1 at observation site O_1 with $\Sigma_{o_1} = \{a, b\}$

Example of Codiagnosability
Local Diagnosers (2)



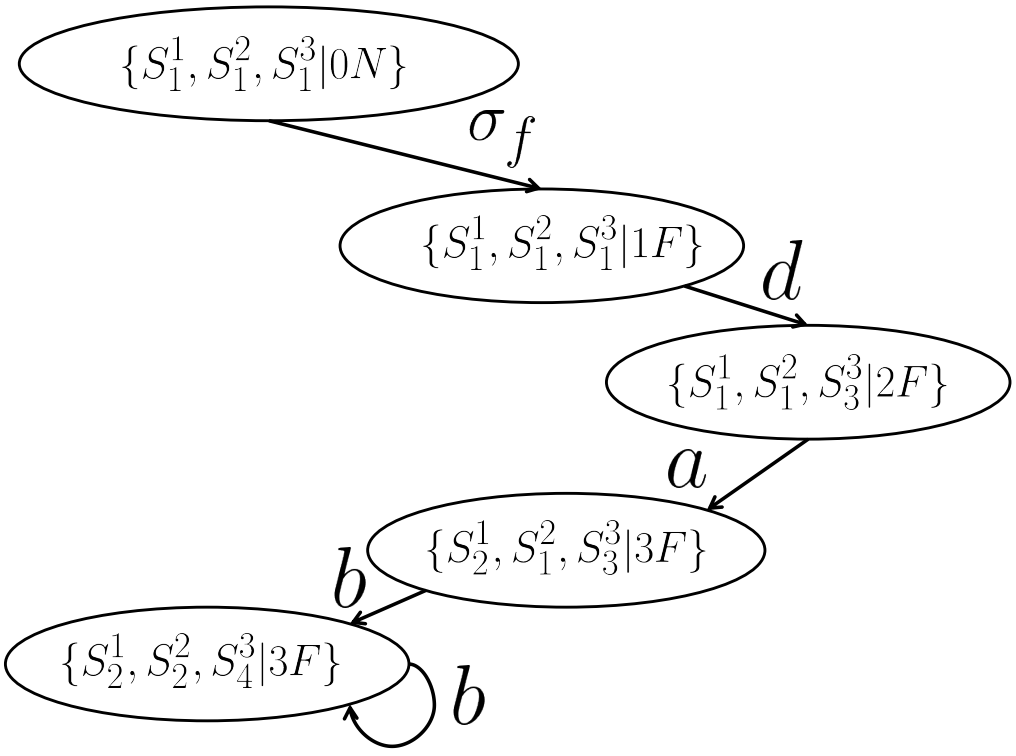
Local diagnoser D_2 at observation site O_2 with $\Sigma_{o_2} = \{b, c\}$

Example of Codiagnosability
Local Diagnosers (3)



Local diagnoser D_3 at observation site O_3 with $\Sigma_{o_3} = \{b, d\}$

Example of Codiagnosability
Parallel Product of Local Diagnosers with System

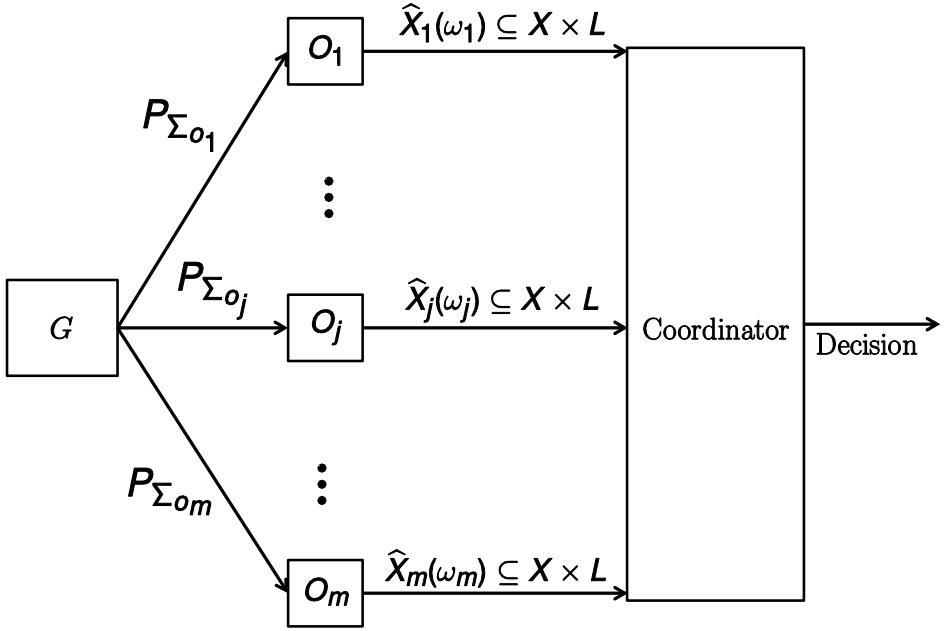


Partial parallel product $D_1||D_2||D_3||G$ for codiagnosability verification.

- Indeterminate cycle at state $\{S_2^1, S_2^2, S_4^3|3F\}$: Each individual local site remains confused since
$$S_2^1 = \{3F, 4N\}\ ,\ S_2^2 = \{3F, 4N\},\ S_4^3 = \{3F, 4N\}$$
- Using parallel product of verifiers results in polynomial complexity in the size of the system (exponential in the number of observation sites)

Fault Diagnosis in Decentralized Settings

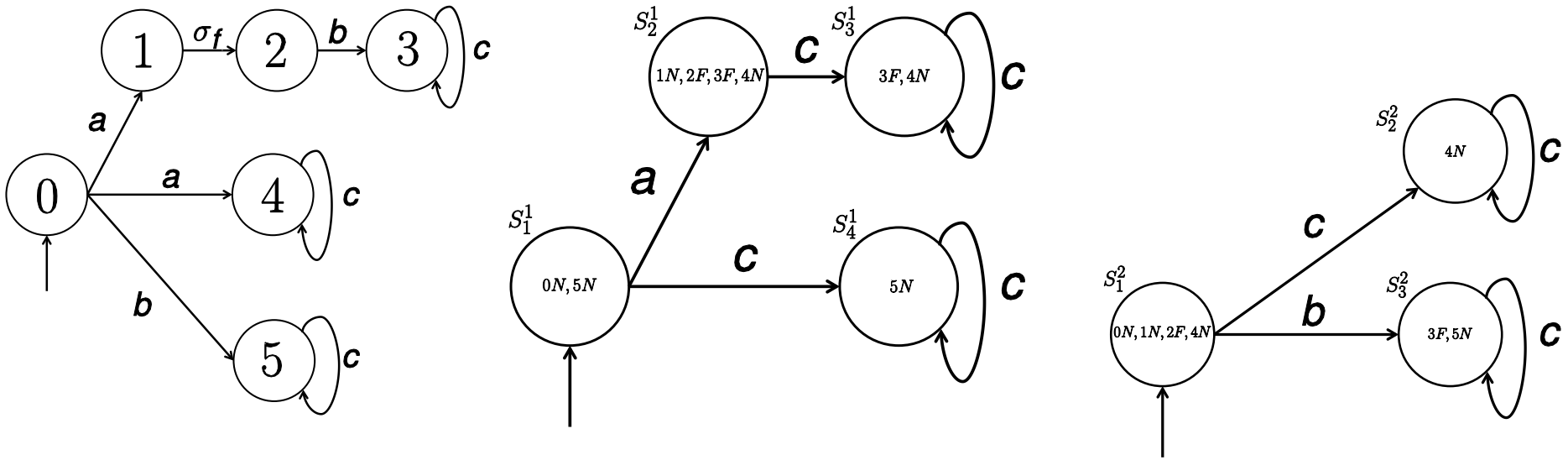
Case 2: Intersection Based Decentralized Diagnosis (IBDD)



- IBDD [Panteli and CNH, 2013]: Send local diagnostic information (state estimates and associated normal/fault conditions)
- Coordinator simply takes intersection (still open loop)
- Problematic cases:
  Indeterminate cycles (of uncertain states) capture system behavior in which *all* diagnosers remain confused *and* coordinator (after taking intersection) also remains confused

Example: Codiagnosability vs IBDD

Sending Decision vs Diagnostic Information



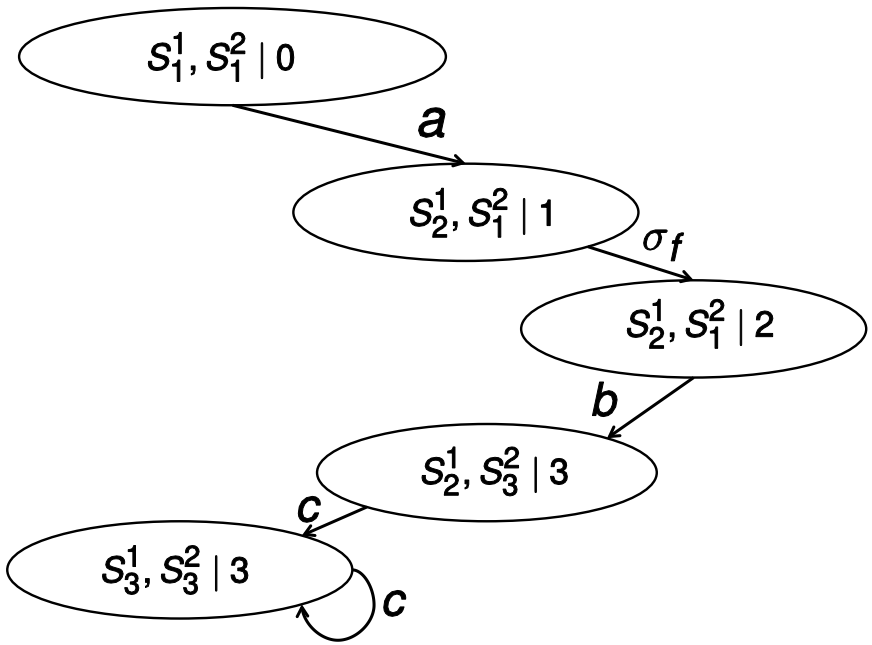
Left: Example system; Middle: Diagnoser D_1; Right: Diagnoser D_2

- $m = 2$, $\Sigma_{o_1} = \{a, c\}$, $\Sigma_{o_2} = \{b, c\}$
- Suppose $s = a\sigma_f bc^n$ occurs; then
  1. Observation site O_1: $\omega_1 = P_{\Sigma_{o_1}}(s) = ac^n$ and $\widehat{X}_1(\omega_1) = \{3F, 4N\}$
  2. Observation site O_2: $\omega_2 = P_{\Sigma_{o_2}}(s) = bc^n$ and $\widehat{X}_2(\omega_2) = \{3F, 5N\}$
  3. Both sites are confused indefinitely (for any n); *not* codiagnosable
  4. IBDD-diagnosable because intersection at coordinator yields

$$\{3F, 4N\} \cap \{3F, 5N\} = \{3F\} \Rightarrow \text{Coordinator detects fault}$$

Verification of Codiagnosability and IBDD

Parallel Product of Local Diagnosers with System

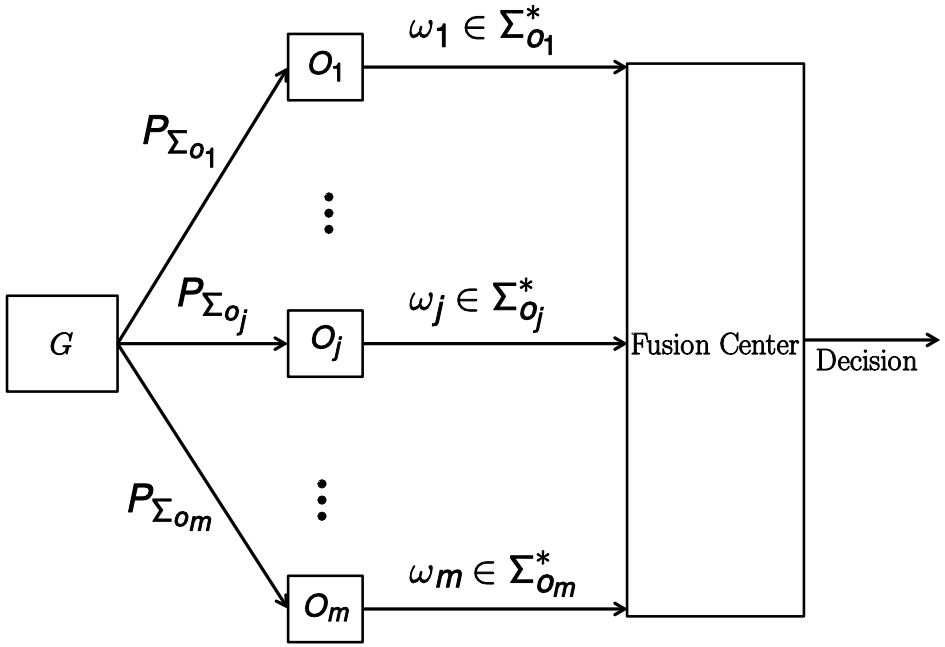


Part of product of local D_1 and D_2, with system G, for verification of codiagnosability and IBDD-diagnosability

- Indeterminate states for codiagnosability: Each individual local site remains confused
  ($S_3^1 = \{3F, 4N\}$ and $S_3^2 = \{3F, 5N\}$ have state estimates with N and F)
- Indeterminate states for IBDD: Intersection of diagnostic information of local sites contains state estimates with both N and F labels
- Verification using verifiers (tracking pairwise uncertainties) has polynomial complexity (in both cases)

Fault Diagnosis in Decentralized Settings

Case 3: Sending Sequences of Observations



- No local processing: send all available information
- Synchronization issues increase processing complexity (at coordinator)
- If no time stamps are available, only partial ordering is possible
  E.g., suppose $m = 2$, $\Sigma_{o_1} = \{\alpha_1, \alpha_2, \gamma\}$, and $\Sigma_{o_2} = \{\beta_1, \beta_2, \gamma\}$

  If $\omega_1 = \alpha_1\gamma\alpha_2$ and $\omega_2 = \beta_1\gamma\beta_2$ then possible orders are:

  $\alpha_1\beta_1\gamma\alpha_2\beta_2, \quad \beta_1\alpha_1\gamma\alpha_2\beta_2, \quad \beta_1\alpha_1\gamma\beta_2\alpha_2, \quad \alpha_1\beta_1\gamma\beta_2\alpha_2$

- Consider $m = 2$ and $\Sigma_{o_1} \cap \Sigma_{o_2} = \emptyset$; also, let $\Sigma_o = \Sigma_{o_1} \cup \Sigma_{o_2}$
- Unknown event sequence $s \in L(G)$ induces
  Available: $\omega_1 = P_{\Sigma_{o_1}}(s) = \alpha_1\alpha_2...\alpha_{L_1}$, where $\alpha_k \in \Sigma_{o_1}$ for $k = 1, 2, ..., L_1$
  Available: $\omega_2 = P_{\Sigma_{o_2}}(s) = \beta_1\beta_2...\beta_{L_2}$, where $\beta_k \in \Sigma_{o_2}$ for $k = 1, 2, ..., L_2$
  Fictitious centralized observer: $\omega_M = P_{\Sigma_o}(s)$
- Set of matching (totally ordered) observation sequences

$$PO(\omega_1, \omega_2) = \{\omega \in \Sigma_o^* \mid P_{\Sigma_j}(\omega) = \omega_j,\ j = 1, 2\}$$

  Note: We have $|PO(\omega_1, \omega_2)| = \frac{(L_1+L_2)!}{L_1!L_2!}$ different matching sequences ω, though some of them may be infeasible (in which case, $\widehat{X}_{\Sigma_o}(\omega) = \emptyset$)
- Direct computation of state estimates expensive

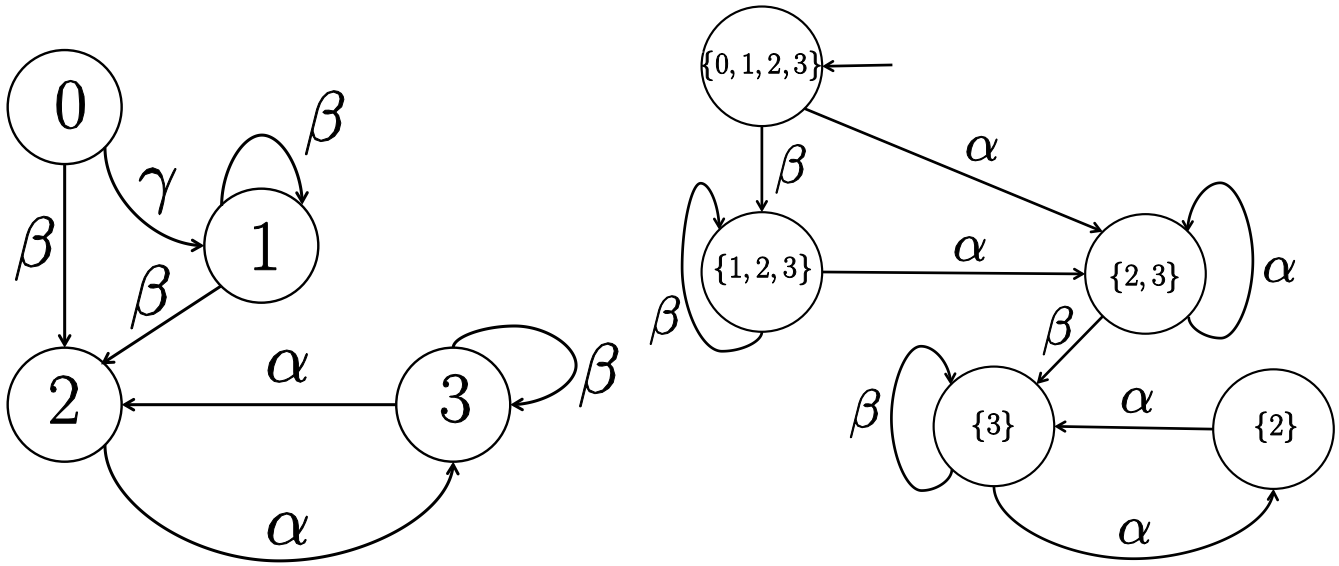
$$\widehat{X}_{po}(\omega_1, \omega_2) = \bigcup_{\omega \in PO(\omega_1, \omega_2)} \widehat{X}_{\Sigma_o}(\omega)$$

- Key observation for efficient recursion [CNH and Seatzu, 2016]

$$PO(\omega_1, \omega_2) = \{t\alpha_{L_1} \mid t \in PO(\omega_1', \omega_2)\} \cup \{t\beta_{L_2} \mid t \in PO(\omega_1, \omega_2')\}$$

  where $\omega_1' = \alpha_1\alpha_2...\alpha_{L_1-1}$ and $\omega_2' = \beta_1\beta_2...\beta_{L_2-1}$

State Estimation under Partial Orders (2)
Example



- Let $m = 2$ with $\Sigma_{o_1} = \{\alpha\}$ and $\Sigma_{o_2} = \{\beta\}$ ($X_0 = X = \{0, 1, 2, 3\}$)
- Consider $s = \alpha\alpha\beta\beta$, so that $\omega_1 = P_1(s) = \alpha\alpha$ and $\omega_2 = P_2(s) = \beta\beta$
- Centralized observer: $\omega_M = P_{\Sigma_o}(s) = s$, leading to $\widehat{X}_{\Sigma_0}(\omega_M) = \{3\}$
- Iterative computation of partial order state estimates ($\widehat{X}_{po}(\omega_1^{1:k_1}, \omega_2^{1:k_2})$)

$\beta\beta$	$\{1, 2, 3\}$	$\{2, 3\}$	$\{2, 3\}$
β	$\{1, 2, 3\}$	$\{2, 3\}$	$\{2, 3\}$
ϵ	$\{0, 1, 2, 3\}$	$\{2, 3\}$	$\{2, 3\}$
k_2/k_1	ϵ	α	$\alpha\alpha$

Comparison of Decentralized State Estimation

Accuracy Tradeoffs Between Different Processing Strategies

- Total Order Estimates: If total order observation sequence ω_M is available (centralized observer or timestamp availability)

$$\widehat{X}_{\Sigma_o}(\omega_M) = \{x \in X \mid \exists x_0 \in X_0, \exists s \in \Sigma^* \text{ s.t. } (P_{\Sigma_o}(s) = \omega_M \wedge x \in \delta(x_0, s))\}$$

- Partial Order Estimates: Process sequences of observations taking into account partial ordering

$$\begin{aligned}\widehat{X}_{po}(\{\omega_1, \omega_2, ..., \omega_m\}) \quad = \quad & \{x \in X \mid \exists x_0 \in X_0, \exists s \in \Sigma^* \text{ s.t.} \\ & (P_{\Sigma_{o_j}}(s) = \omega_j, j = 1, 2, ..., m, \wedge x \in \delta(x_0, s))\}\end{aligned}$$

- Intersection-Based Processing: Obtain sets of state estimates locally at each site and take their intersection

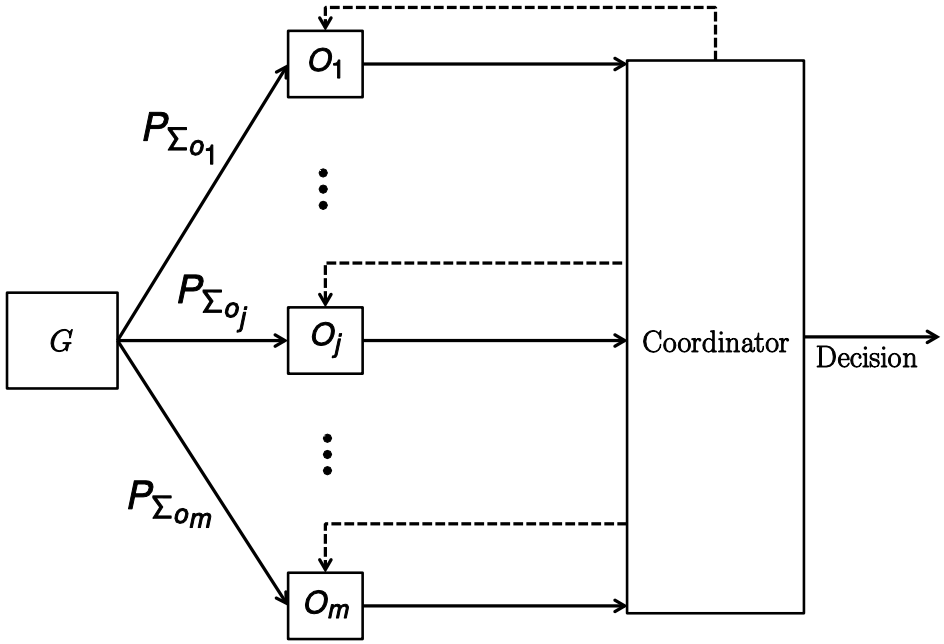
$$\widehat{X}_{ib}(\{\omega_1, \omega_2, ...\omega_m\}) = \bigcap_{j=1}^{m} \widehat{X}_{\Sigma_{o_j}}(\omega_j) \text{ where}$$

$$\widehat{X}_{\Sigma_{o_j}}(\omega_j) = \{x \in X \mid \exists x_0 \in X_0, \exists s \in \Sigma^* \text{ s.t. } (P_{\Sigma_{o_j}}(s) = \omega_j \wedge x \in \delta(x_0, s))\}$$

- Tradeoff in accuracy:

$$\widehat{X}_{\Sigma_o}(\omega_M) \subseteq \widehat{X}_{po}(\{\omega_1, \omega_2, ..., \omega_m\}) \subseteq \widehat{X}_{ib}(\{\omega_1, \omega_2, ...\omega_m\})$$

Fault Diagnosis in Distributed Settings
Closed Loop Architectures with Coordinator



- When/what information is sent from an observation site to the coordinator?
- When/what information is sent back from the coordinator?
- Relates to dynamic sensor activation but local decisions need to rely on locally available information
- One option: Simultaneous feedback from the coordinator to all sites [Keroglou and CNH, 2018]

- Future work and open questions in NFA setting:
  1. Alternative synchronization strategies
     a. Develop optimal strategies (e.g., minimize communication overhead while ensuring detectability/diagnosability)
     b. Characterize/generalize polynomially verifiable strategies
  2. Role of modularity and/or other system structure
  3. Resiliency
     a. Syncrhonization issues (delays, asynchronous operation)
     b. Tolerance to packet drops, errors in communication exchanges, faulty/malicious sites

- Extensions to other DES:
  1. Petri net models (and other purely event-driven systems)
  2. Stochastic systems (introduce and verify stochastic notions of diagnosability/observability)

- Extensions to Cyberphysical systems:
  1. Intersection-based protocols?
  2. Triggering strategies for synchronization?
  3. Which protocols lead to polynomially verifiable properties?

References

- **[Cassandras and Lafortune, 2008]** C. G. Cassandras and S. Lafortune. Introduction to discrete event systems. Springer Science & Business Media, 2009.
- **[CNH, 2018]** C. N. Hadjicostis. Model-Based Estimation and Inference in Discrete Event Systems, Draft available at http://www.eng.ucy.ac.cy/hadjicostis/JournalPapers/efddraftECE800.pdf
- **[Lin and Wonham, 1988]** F. Lin and and W. M. Wonham, “On observability of discrete-event systems," Information Sciences, 44.3: 173–198.
- **[Ozveren and Willsky, 1990]** C. M. Ozveren and A. S. Willsky, “Observability of discrete event dynamic systems," IEEE Transactions on Automatic Control, 35.7: 797–806.
- **[Caines et al, 1991]** P. E. Caines, R. Greiner, and S. Wang, “Classical and logic-based dynamic observers for finite automata," IMA Journal of Mathematical Control and Information, 8.1: 45–80.

References (2)

- **[Shu et al, 2007]** S. Shu, F. Lin, and H. Ying, “Detectability of discrete event systems," IEEE Transactions on Automatic Control, 52.12: 2356–2359.
- **[Shu and Lin, 2010]** S. Shu and F. Lin, “Detectability of discrete event systems with dynamic event observation," Systems and Control Letters, 59.1: 9–17.
- **[Sampath et al, 1995]** M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Diagnosability of discrete-event systems," IEEE Transactions on Automatic Control, 40.9: 1555–1575.
- **[Jiang et al, 2001]** S. Jiang, Z. Huang, V. Chandra, and R. Kumar, “A polynomial algorithm for testing diagnosability of discrete-event systems," IEEE Transactions on Automatic Control, 46.8: 1318–1321.
- **[Yoo and Lafortune, 2002]** T.-S. Yoo and S. Lafortune, “Polynomial-time verification of diagnosability of partially observed discrete-event systems," IEEE Transactions on Automatic Control, 47.9: 1491–1495.

References (3)

- **[Saboori and CNH, 2007]** A. Saboori and C. N. Hadjicostis, “Notions of security and opacity in discrete event systems," Proceedings of 46th IEEE Conference on Decision and Control.
- **[Saboori and CNH, 2011]** A. Saboori and C. N. Hadjicostis, “Verification of K-step opacity and analysis of its complexity," IEEE Transactions on Automation Science and Engineering, 8.3: 549–559.
- **[Saboori and CNH, 2008]** A. Saboori and C. N. Hadjicostis, “Verification of initial-state opacity in security applications of DES," Proceedings of 9th International Workshop on Discrete Event Systems.
- **[Badouel et al, 2006]** E. Badouel, M. Bednarczyk, A. Borzyszkowski, B. Caillaud, and P. Darondeau, “Concurrent secrets," Discrete Event Dynamic Systems, 17.4: 425–446.
- **[Wu and Lafortune, 2014]** Y.-C. Wu and S. Lafortune, “Synthesis of insertion functions for enforcement of opacity security properties," Automatica, 50.5: 1336–1348.

References (4)

- **[Debouk et al, 2000]** R. Debouk, S. Lafortune, and D. Teneketzis, “Coordinated decentralized protocols for failure diagnosis of discrete event systems," Discrete Event Dynamic Systems 10.1-2: 33–86.
- **[Wang et al, 2007]** Y. Wang, T.-S. Yoo, and S. Lafortune, “Diagnosis of discrete event systems using decentralized architectures," Discrete Event Dynamic Systems 17.2: 233–263.
- **[Qiu and Kumar, 2006]** W. Qiu and R. Kumar, “Decentralized failure diagnosis of discrete event systems," IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 36.2: 384–395.
- **[Panteli and CNH, 2013]** M. Panteli and C. N. Hadjicostis, “Intersection based decentralized diagnosis: Implementation and verification," Proceedings of 52nd IEEE Conference on Decision and Control.
- **[CNH and Seatzu, 2016]** C. N. Hadjicostis and C. Seatzu, “Decentralized state estimation in discrete event systems under partially ordered observation sequences," Proceedings of 13th International Workshop on Discrete Event Systems.
- **[Keroglou and CNH, 2018]** C. Keroglou and C. N. Hadjicostis, “Distributed fault diagnosis in discrete event systems via set intersection refinements," IEEE Transactions on Automatic Control.