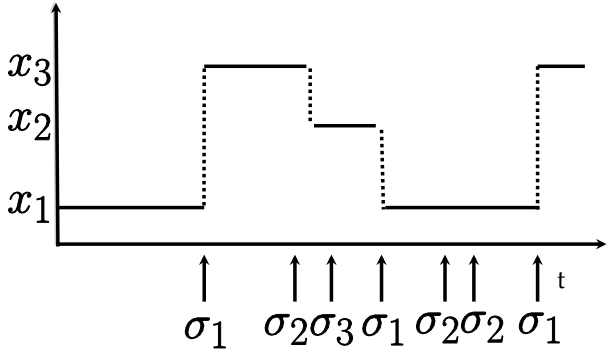
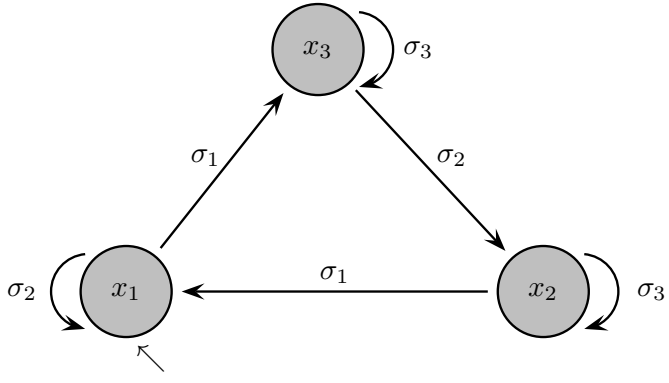
Notions of Opacity for Privacy and Security in Discrete Event Systems

Christoforos N. Hadjicostis

Department of Electrical and Computer Engineering
University of Cyprus

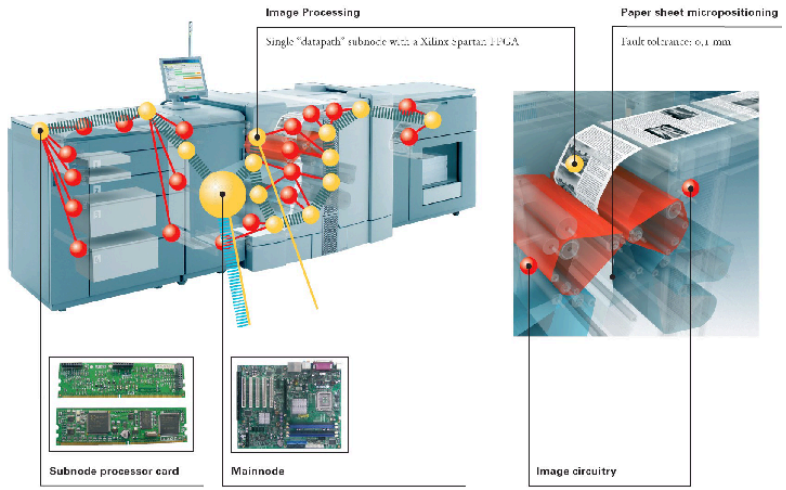
- Event-driven dynamics [Cassandras and Lafortune, 2008]
  1. Numerous digital/cyberphysical systems are naturally event-driven (e.g., asynchronous distributed systems) or even exclusively event-driven (e.g., communication protocols)
  2. Event-driven sampling may be a design choice (e.g., [Branicky and Phillips, 2000], [Astrom and Bernhardsson, 2002], [Tabuada, 2007], [Dimarogonas et al, 2012], [Cassandras, 2015])
- Discrete state space (typically)
  1. Finite (e.g., finite automata) or infinite (e.g., unbounded Petri nets)
  2. Extensions to timed/stochastic/hybrid models (e.g., hybrid automata, continuous Petri nets)
  3. Diverse levels of abstraction: Logical, stochastic, hierarchical, ...





- “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 (e.g., expensive) or unreliable sensors, different/complex modes of operation (e.g., monitoring vs testing), complexity of verification process



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

- New Features and Characteristics:
  - Distributivity/Modularity: Multiple interacting (sub)systems, local observers and controllers
  - Processing: Local vs global, exchange of information
  - Optimization: Collaborative vs antagonistic strategies
  - Communication: Network delays, packet drops, synchronization
- Privacy and Security Challenges:
  - Shared (non-dedicated) communication infrastructures
  - Compromised components (e.g., sensors or actuators)
  - Curious or malicious actors (e.g., intruders)

Opacity in Discrete Event Systems
General (Behavioral) Description

- Certain critical system behavior deemed secret [Bryans et al., 2004] (described by predicate that evaluates to true or false)
- Curious observers (or passive intruders) are assumed to have
  1. Knowledge of a (possibly partial) model of the system
  2. Partial access to activity (observations) generated by the system
- Curious observers do not interfere with system operation in other ways (subsequent workshop talks address the effects of malicious intruders)
- Opacity requires the secret system behavior to remain opaque (uncertain) to passive intruders, under *all* system behavior
- Opaque system implies that the curious observer

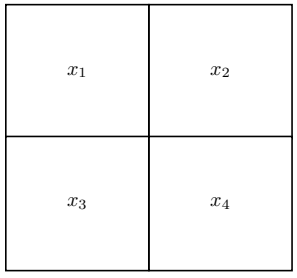
  never establishes that the predicate describing secret behavior is true

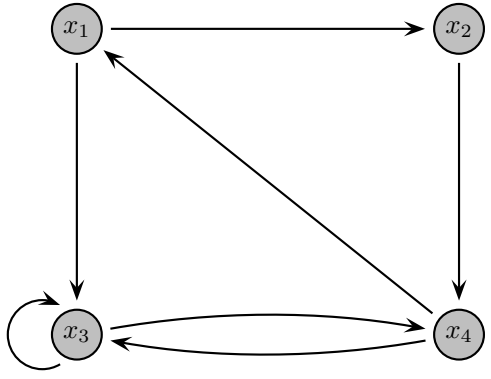
- Probabilistic extensions (more generally, ways of quantifying opacity) are also possible [Saboori and CNH, 2014]

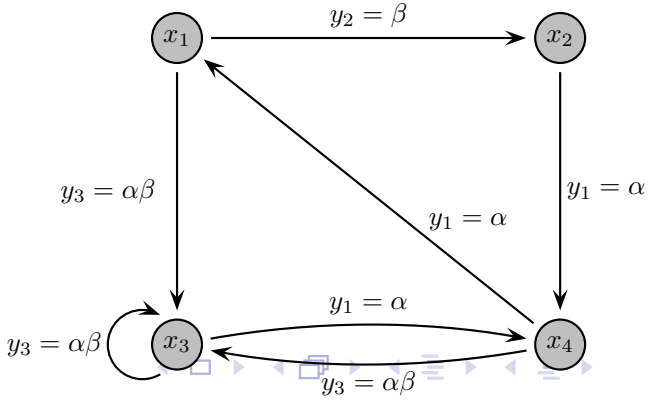
- Motivating applications include assessment of
  - Monitoring limitations in sensor networks
    [Dubreil et al., 2010], [Saboori and CNH, 2011]
  - Encryption mechanisms based on pseudorandom generators
    [Saboori, 2011]
  - Protocols for privacy-preserving location-based services
    [Wu et al., 2014]
- Several existing security notions, such as anonymity and noninterference, can also be described using opacity formulations

Motivating Example I
Monitoring Limitations in Sensor Networks

- Vehicle moves on a two-dimensional grid in which a number of sensors is deployed
- State of the vehicle corresponds to cell number of its location $\Rightarrow$ Vehicle trajectory corresponds to state trajectory
- Kinematic Model: Automaton describing limitations on vehicle movements due to physical obstacles on the grid or other logical constraints or rules
- Enhanced Kinematic Model: Assign observation σ to all transitions that end in a cell covered by sensor σ





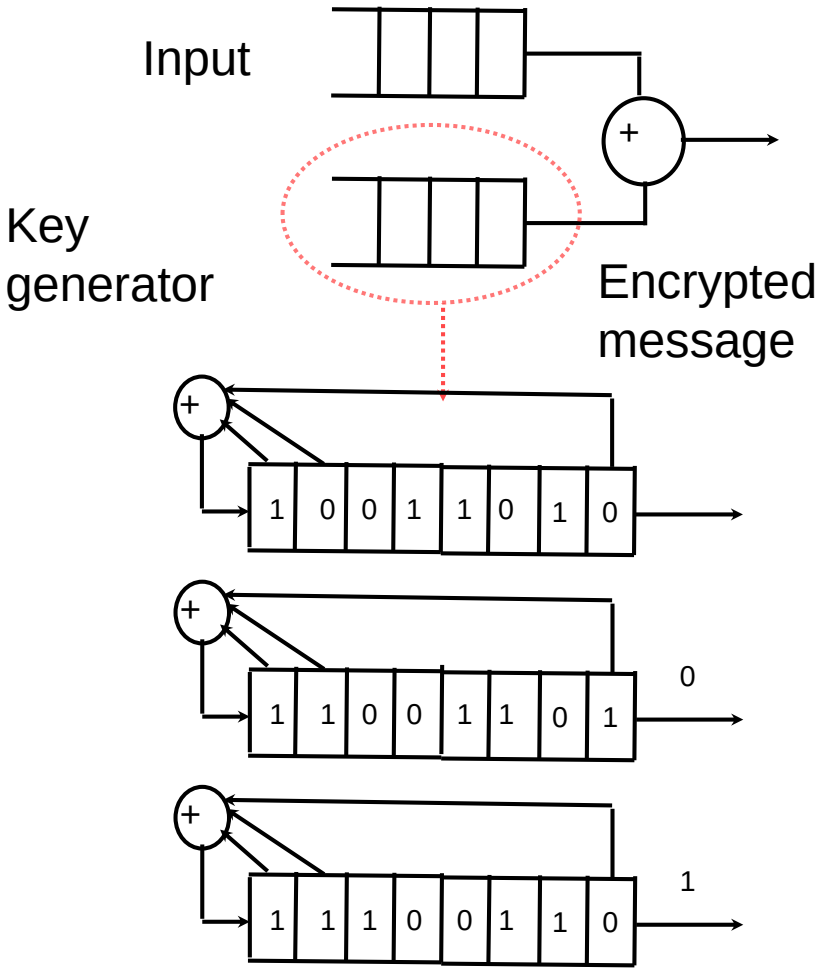


Motivating Example I
Examples of Privacy/Security Concerns

- Can the origin of the vehicle be identified via observations from the sensor network (for all or some trajectories)?
- For such trajectories, how long (in the best/worst case) does it take for the sensor network to identify the origin?
- How fast does the number of consistent trajectories increase in terms of the length of the observation sequence?
- What sensor placement results in the tightest estimate of the vehicle state (either at present or at some point in the past) in the shortest time?
- What extra restrictions can we impose on vehicle movements in order to improve or impair our ability to localize the vehicle (i.e., perform state estimation) without changing sensor configuration?
- Assuming that some statistics about vehicle movements are known a priori, what is the most probable state of the vehicle along the observation?

Motivating Example II
Limitations of Encryption using Pseudorandom Generators

- Stream cipher: combines (usually through an XOR operation) plain text bits with a stream of keys
- LFSR-based stream cipher: Linear Feedback Shift Register creates a pseudorandom stream of keys
- An intruder can insert input bits and observe the encrypted message in an effort to obtain information about the system



8-bit LFSR with tap sequence 0,1,7

Motivating Example II
Examples of Privacy/Security Concerns

- Is there an initial key (state) for which there exists an input sequence that reveals that key?
- If there is such a key, how long does it take for the intruder to detect it?
- Is there a (bad) sequence of key resets that aids the intruder in identifying the current (or previous) key faster?
- Assuming that some statistics about the initial key are known, what input sequence will reveal the key with the highest probability?

Observability Related Challenges

DES Models and Property Verification [CNH, 2020]

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, ...

Sources of Uncertainty

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

Opacity Specific Challenges:

1. Online algorithms for state estimation and event inference
2. Verification and complexity of opacity notions of interest
3. Automated tools for verifying/enforcing properties of interest

Recent book: CNH, *Estimation and Inference in Discrete Event Systems: A Model-Based Approach with Finite Automata*, Springer, 2020

Talk Outline

- Introduction and Motivation
  1. Opacity as a privacy notion
  2. Motivating examples
- Observation Models; Language-Based and State-Based Opacity
- Current-State Opacity and its Verification
  1. Current-state estimation
  2. Formal definition of current-state opacity
  3. Verification using observer (current-state estimator)
- Initial-State Opacity and its Verification
  1. Initial-state estimation
  2. Formal definition of initial-state opacity
  3. Verification using initial-state estimator
- Other State-Based Notions of Opacity
- Ongoing Research and Challenges

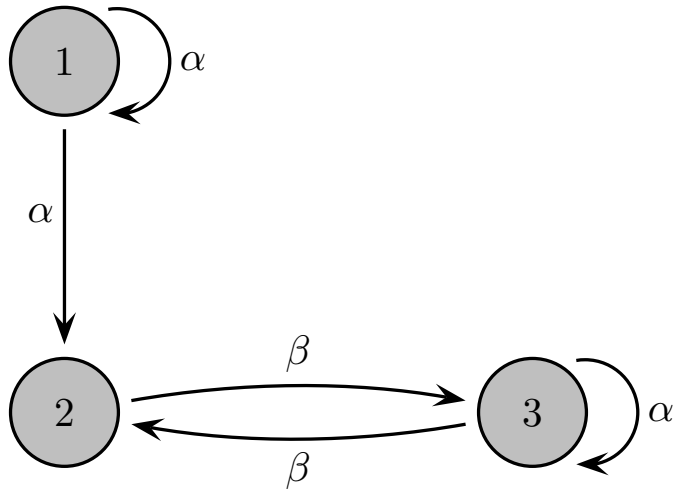
Nondeterministic Finite Automaton (NFA)
Notation

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

- X is the set of states
- Σ is the set of events
- $\delta : X \times \Sigma \to 2^X$
  nondeterministic transitions
  (Deterministic if $|\delta(x, \sigma)| \leq 1$
  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 $x \in X$, $s \in \Sigma^*$, $\sigma \in \Sigma$

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 information, time stamps)

Notions of Opacity
Language-Based Opacity [Lin, 2011]

- Given NFA $G = (X, \Sigma, \delta, X_0)$ with set of observable events Σ_o ($\Sigma_o \subseteq \Sigma$)
- Curious observer with
  (i) knowledge of system model G and
  (ii) access to the sequence of observations $\omega = P(s)$ (generated in response to sequence of events s, $s \in L(G)$, acting in the system)
- Language-Based Opacity: G is said to be opaque with respect to the secret language L_S ($L_S \subset L(G)$) if

  $\forall s \in L_S$, we can find $t \in (L(G) \setminus L_S)$, such that $P(s) = P(t)$
- Nomeclature: L_S is the secret language, whereas $L_{NS} = L(G) \setminus L_S$ is the non-secret language; intuitively, the curious observer should never know that the system has executed a secret sequence of events
- Extensions:
  - Arbitrary non-secret language L_{NS} (not necessarily $L(G) \setminus L_S$)
  - Weak language-based opacity: we can find $s \in L_S$ and $t \in (L(G) \setminus L_S)$, such that $P(s) = P(t)$

Notions of Opacity

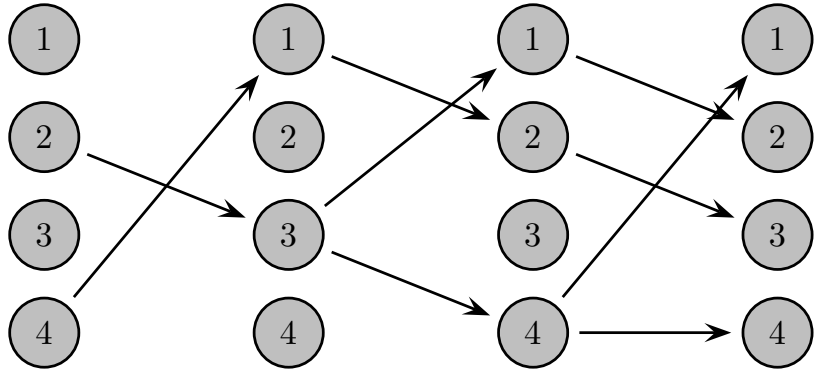
State-Based Opacity [Saboori and CNH, 2007; 2009; 2011; 2013]

- Given NFA $G = (X, \Sigma, \delta, X_0)$ with set of observable events Σ_o ($\Sigma_o \subseteq \Sigma$)
- Curious observer with
  (i) knowledge of system model G and
  (ii) access to the sequence of observations $\omega = P(s)$ (generated in response to sequence of events s, $s \in L(G)$, acting in the system)
- State-Based Opacity: G is said to be (current-state) opaque with respect to the set of secret states S ($S \subset X$) if

  $\forall s \in L(G)$, we can find $t \in P^{-1}(s)$ such that $\delta(X_0, t) \cap (X \setminus S) \neq \emptyset$
- Nomeclature: S is the set of secret states, whereas $NS = X \setminus S$ is the set of non-secret states; intuitively, the curious observer should never know with certainty that the system is in a secret state
- Extensions:
  - Arbitrary set of non-secret states NS (not necessarily $X \setminus S$)
  - Weak state-based opacity: there exists $s \in L_S$, for which we can find $t \in P^{-1}(s)$ such that $\delta(X_0, t) \cap (X - S) \neq \emptyset$

State-Based Opacity
Variations Based on Point in Time

- Current-state opacity: Entrance of current state to set of secret states S remains opaque
- Initial-state opacity: Membership of initial state to set of secret states S_0 remains opaque during system operation
- Other opacity notions allow refinement of the observer estimate based on a subsequently observed sequence of observations (smoothing); these include K-step opacity, infinite-step opacity, etc.
- Key challenge: To verify (strong) opacity, we need to check these conditions for all possible sequences of observations



- $X = \{1, 2, 3, 4\}$
- State trajectories matching a sequence of 3 observations
- System is not initial-state opaque for $S_0 = \{2, 4\}$
- System is not 2-step state opaque for $S = \{1, 3\}$

Verification of State-Based Opacity

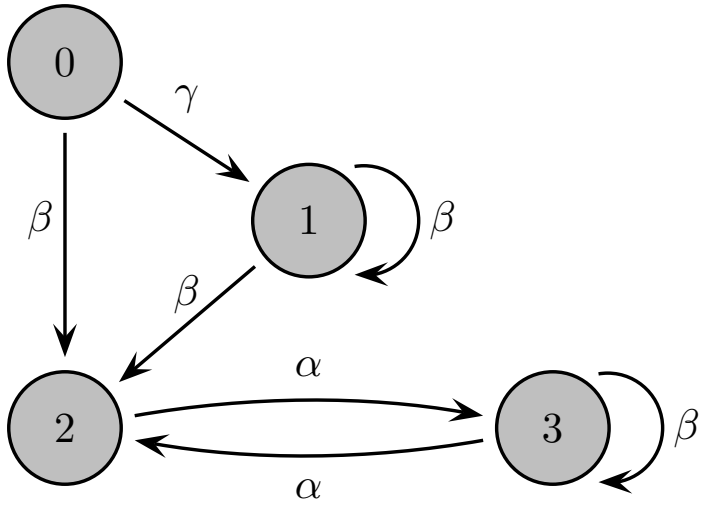
State Estimator Constructions [CNH, 2020]

- Given NFA $G = (X, \Sigma, \delta, X_0)$ with set of observable events Σ_o ($\Sigma_o \subseteq \Sigma$) and a curious observer
- For regular languages, language-based and state-based opacity are equivalent within polynomial reduction [Wu and Lafortune, 2013]
- Verification of state-based opacity relies on different estimators
  - Current-state estimator (or observer)
  - Initial-state estimator
  - K-step delayed estimator
  - Two-way observer
- Complexity depends on estimator complexity (typically, at least exponential in the size of G)
- More efficient verification may be possible in some cases

Talk Outline

- Introduction and Motivation
  1. Opacity as a privacy notion
  2. Motivating examples
- Observation Models; Language-Based and State-Based Opacity
- Current-State Opacity and its Verification
  1. Current-state estimation
  2. Formal definition of current-state opacity
  3. Verification using observer (current-state estimator)
- Initial-State Opacity and its Verification
  1. Initial-state estimation
  2. Formal definition of initial-state opacity
  3. Verification using initial-state estimator
- Other State-Based Notions of Opacity
- Ongoing Research and Challenges

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

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

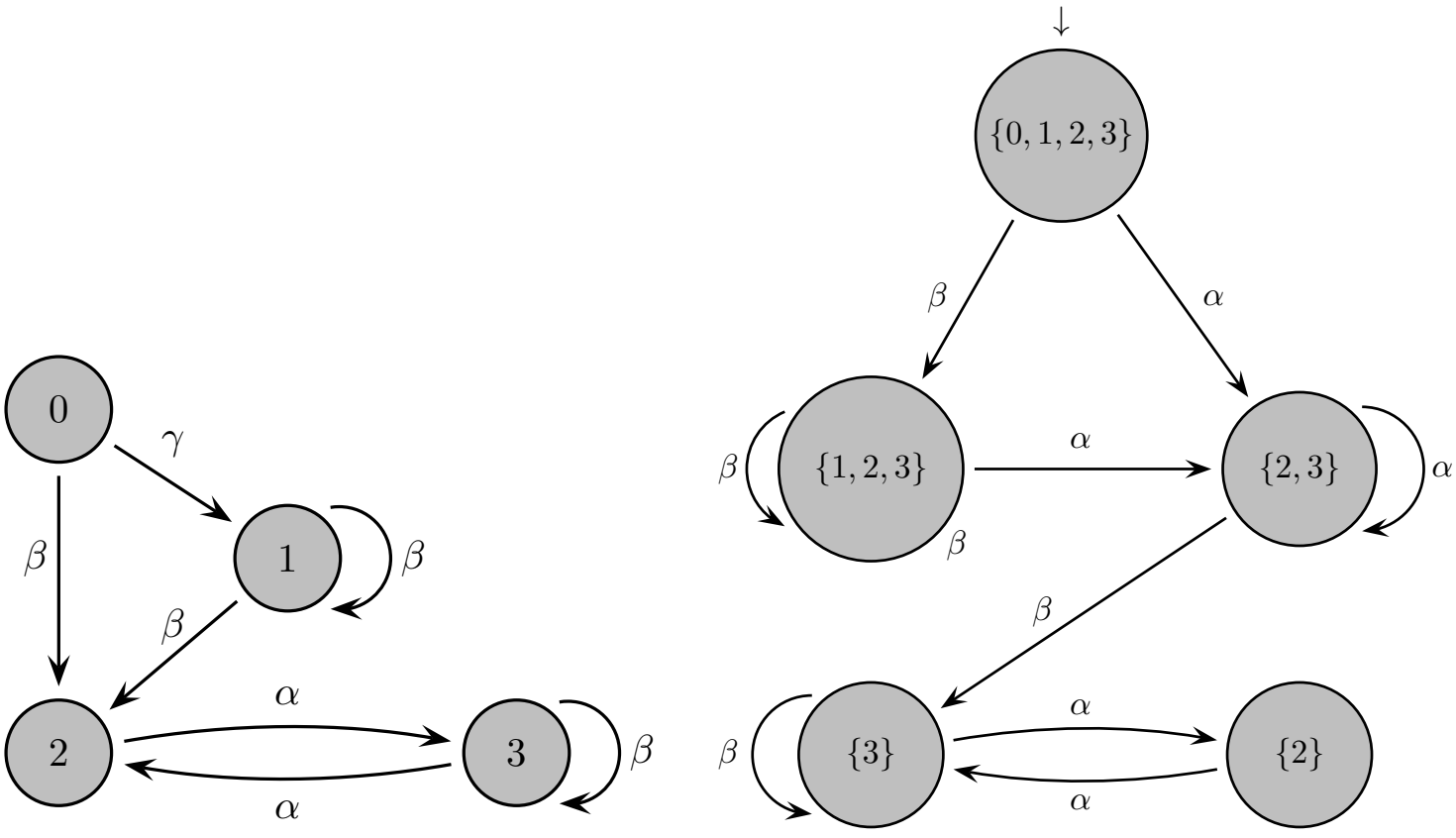
- Given $\omega = \sigma_{i_1}\sigma_{i_2}...\sigma_{i_k} \in \Sigma_o^*$ and $\sigma_{i_{k+1}} \in \Sigma_o$, we can obtain the set $\widehat{X}$ recursively as

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

- Need: Knowledge of system model or $R(X', \sigma_o)$ for $X' \subseteq X$ and $\sigma_o \in \Sigma_o$

Observer (Current-State Estimator) Construction

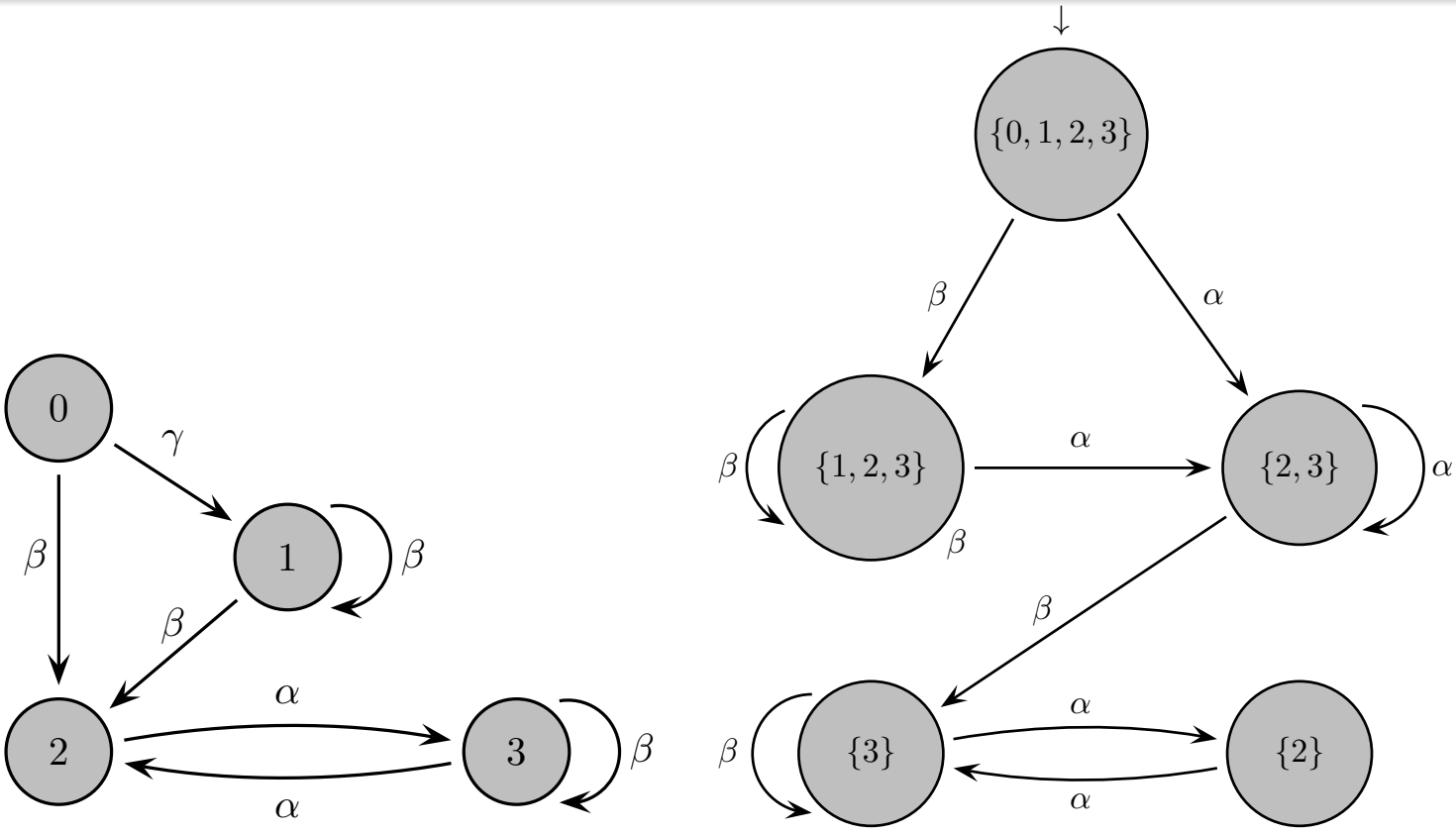
Tracking State Estimates Following *Any* Sequence of Observations



- Sequence of observations $\alpha\beta\alpha$ leads us to state $\{2\}$ (as seen earlier)

Observer Limitations

No Tracking of State Sequences Following Sequences of Observations



- Sequences of states matching $\beta\beta$ (which leads to $\{1,2,3\}$)
  $0 \to 1 \to 1 \to 1$
  $1 \to 1 \to 1$
  $0 \to 1 \to 1 \to 2$
  $1 \to 1 \to 2$
  $3 \to 3 \to 3$
- Current-state estimator does not track state sequences

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 given 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)

Current-State Opacity

Formal Definition and Verification [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 ($S \subseteq X$)

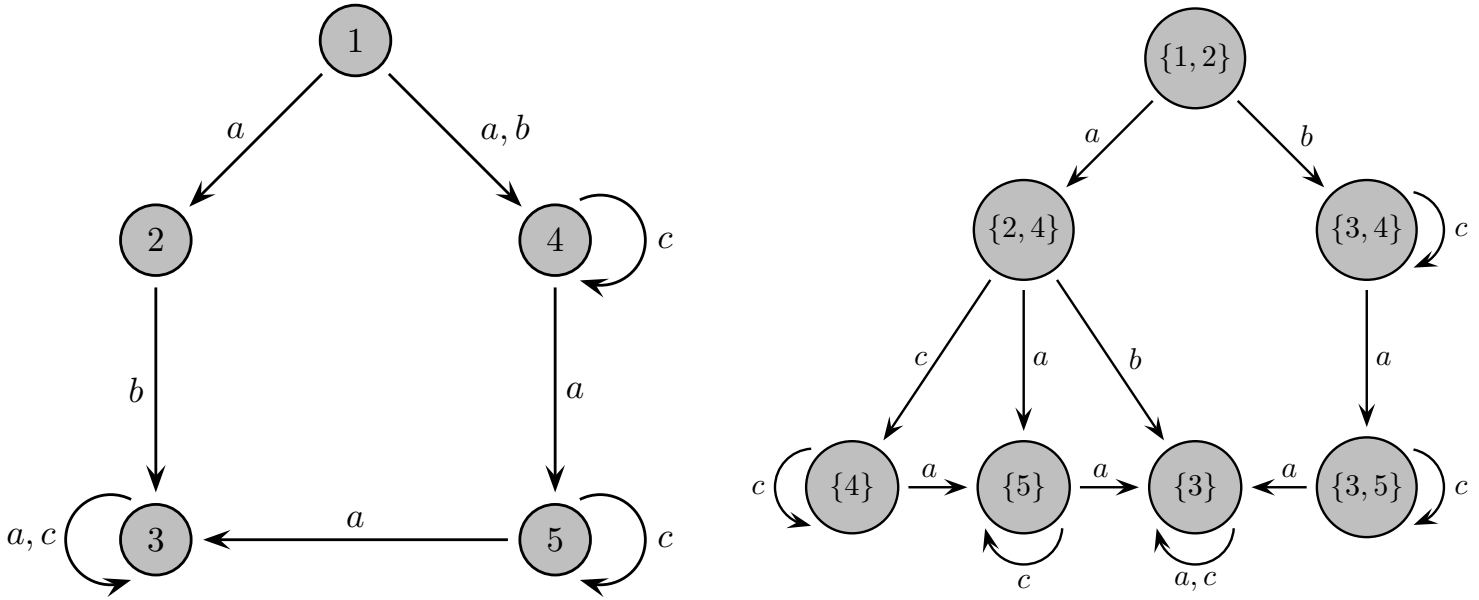
- Current-state opacity requires that an external observer can never be certain that system state is within the set of secret states S
  [At least one state outside S is possible; relates to "possible innocence" in anonymity protocols]

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

- Verification using an observer: G is current-state opaque with respect to a set of secret states S, $S \subset X$, if and only if

$$\forall q_{obs} \in Q_{obs}, \text{ we have } q_{obs} \cap (X \setminus S) \neq \emptyset$$

Verification of Current-State Opacity
Example



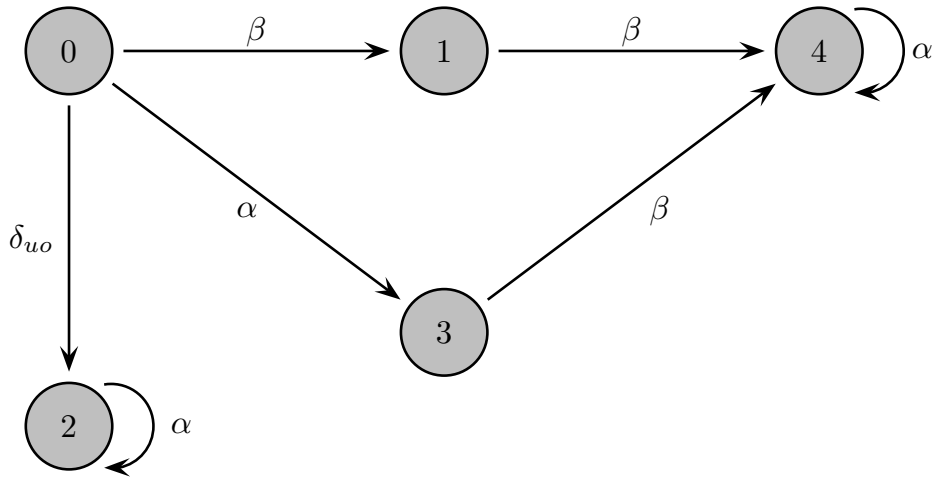
- $X_0 = \{1, 2\}$, $\Sigma_o = \Sigma = \{a, b, c\}$
- Not current-state opaque wrt $S = \{3\}$ or $S = \{4\}$ or $S = \{5\}$
- Current-state opaque wrt $S = \{1\}$ or $S = \{2\}$
- Consider $S = \{4\}$: sequence of observations ac reveals that current state is 4, however, remaining behavior is not opacity violating
- Enforcement of opacity via (i) appropriate control (supervisory control) or (ii) obfuscation of observations (sensor manipulation)

Talk Outline

- Introduction and Motivation
  1. Opacity as a privacy notion
  2. Motivating examples
- Observation Models; Language-Based and State-Based Opacity
- Current-State Opacity and its Verification
  1. Current-state estimation
  2. Formal definition of current-state opacity
  3. Verification using observer (current-state estimator)
- Initial-State Opacity and its Verification
  1. Initial-state estimation
  2. Formal definition of initial-state opacity
  3. Verification using initial-state estimator
- Other State-Based Notions of Opacity
- Ongoing Research and Challenges

Initial State Estimation

Example of Recursive Computation of Possible Initial States



- Consider NFA G with $X_0 = \{0, 1, 2, 3, 4\}$ and $\Sigma_o = \{\alpha, \beta\}$, $\Sigma_u = \{\delta_{uo}\}$
- 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

$$\left\{ \begin{array}{c} (0,0) \\ (0,2) \\ (1,1) \\ (2,2) \\ (3,3) \\ (4,4) \end{array} \right\} \xrightarrow{\alpha} \left\{ \begin{array}{c} (0,2) \\ (0,3) \\ (2,2) \\ (4,4) \end{array} \right\} \xrightarrow{\beta} \{ \ (0,4) \ \} \xrightarrow{\alpha} \{ \ (0,4) \ \}$$

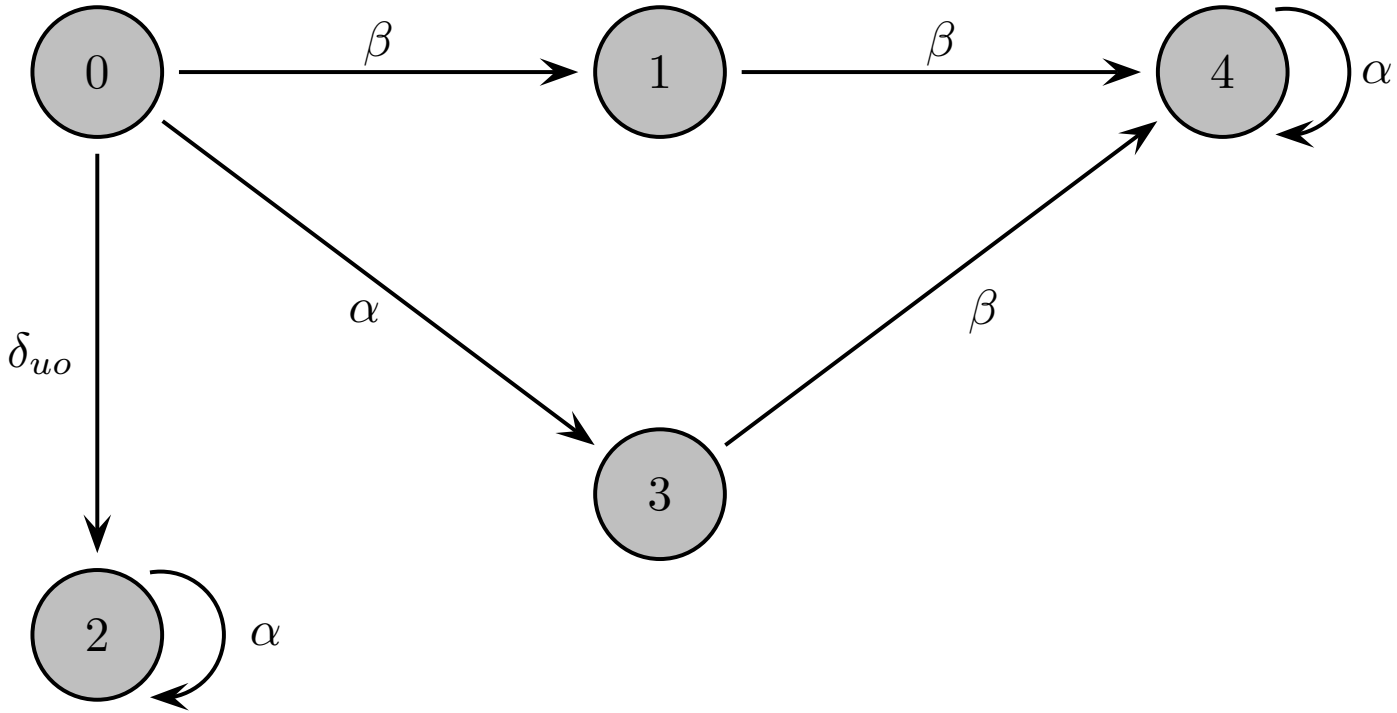
Initial-State Estimation Example

Tracking Initial State Estimates Following Sequences of Observations

Observation: Nothing

Matching Pairs of (Initial, Current) States:
$\{(0,0),(0,2),(1,1),(2,2),(3,3),(4,4)\}$

Initial State Estimate: $\{0,1,2,3,4\}$



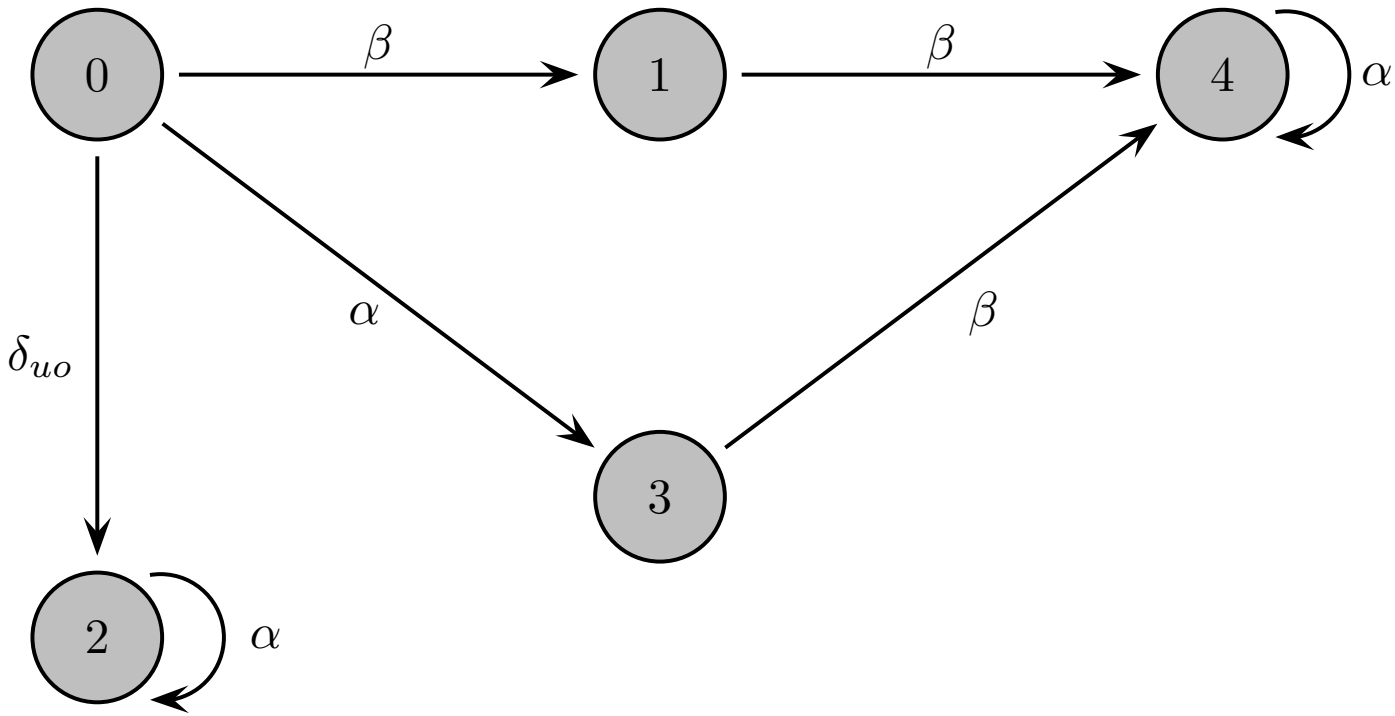
Initial-State Estimation Example (2)

Tracking Initial State Estimates Following Sequences of Observations

Observation: β

Matching Pairs of (Initial, Current) States: $\{(0,1),(1,4),(3,4)\}$

Initial State Estimate: $\{0,1,3\}$



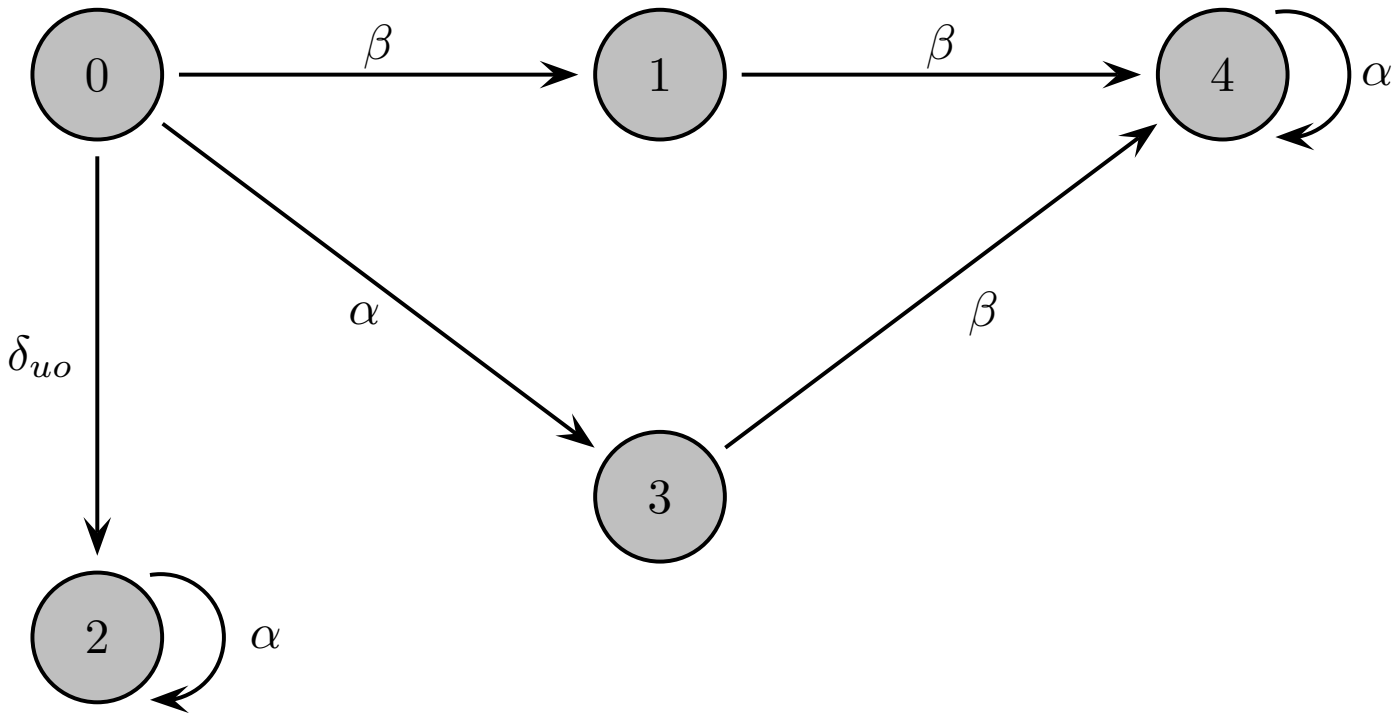
Initial-State Estimation Example (3)

Tracking Initial State Estimates Following Sequences of Observations

Observation: $\beta\beta$

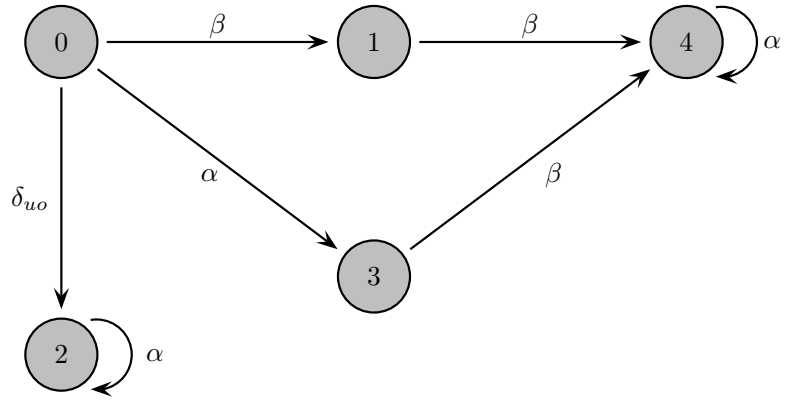
Matching Pairs of (Initial, Current) States: $\{(0, 4)\}$

Initial State Estimate: $\{0\}$



System specifications:

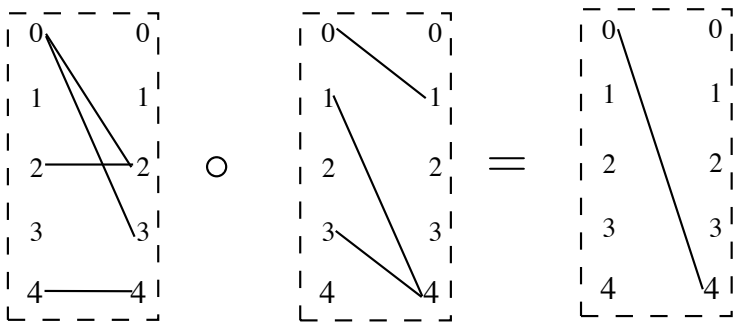
- $X_0 = X = \{0, 1, 2, 3, 4\}$
- $\Sigma = \{\alpha, \beta, \delta_{uo}\}$
- $\Sigma_u = \{\delta_{uo}\}$



Induced State Mappings:

$m_\alpha = \{(0, 2), (0, 3), (2, 2), (4, 4)\}$

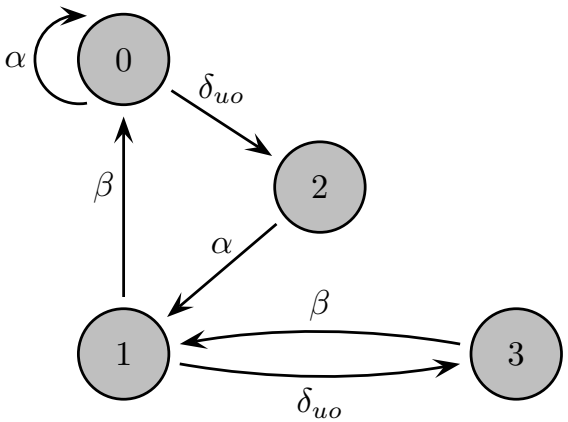
$m_\beta = \{(0, 1), (1, 4), (3, 4)\}$

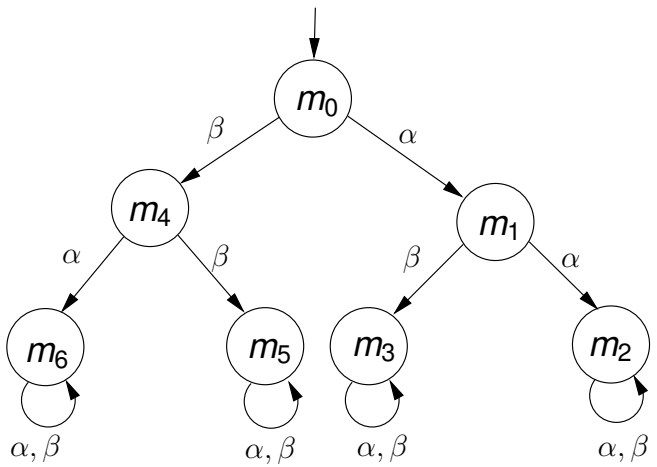


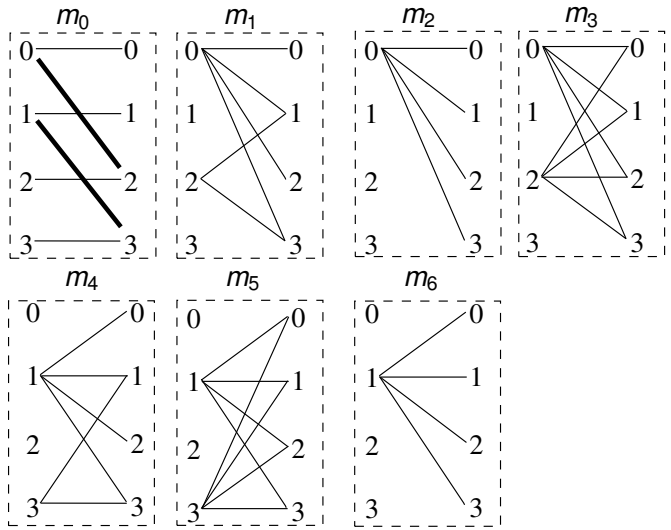
$m_{\alpha\beta}$ can be obtained via appropriate composition of m_α and m_β

$m_{\alpha\beta} = \{(x_{i\alpha}, x_{c\beta}) \mid \exists x_{c\alpha} = x_{i\beta} \text{ s.t. } (x_{i\alpha}, x_{c\alpha}) \in m_\alpha \text{ and } (x_{i\beta}, x_{c\beta}) \in m_\beta\}$

Initial-State Estimator Construction
Example







- ISE has $O(2^{N^2})$ worst-case complexity ($N = |X|$)
- Can be used to determine both initial and current state
  E.g., if $\beta\alpha(\alpha+\beta)^*$ is observed the initial state was 1, whereas current state could be any state in X
- Property: Refinement of set of possible initial states as more observations become available

Formal Construction of Initial-State Estimator

Tracking Pairs of Starting and Final States [Saboori and CNH, 2009; 2013]

- Given: NFA $G = (X, \Sigma, \delta, X_0)$ with observable events Σ_o ($\Sigma_o \subseteq \Sigma$)
- Let $m_{\sigma_o} \subseteq X \times X$ denote the induced state mapping associated with σ_o
- Initial-State Estimator: *Deterministic* finite automaton $G_{iobs} = (Q_{iobs}, \Sigma_o, \delta_{iobs}, Q_{0,iobs})$ constructed as follows:
  1. $Q_{iobs} \subseteq 2^{X \times X}$, i.e., each initial-state estimator state $q_{iobs} \in Q_{iobs}$ is associated with a unique subset of pairs of states of the form (x_i, x_c) where x_i is a possible initial state and x_c is a corresponding possible current state
  2. Initial state is $Q_{0,iobs} = \cup_{x_0 \in X_0} \{\{x_0\} \times R(x_0, \epsilon)\}$ (each $x_0 \in X_0$ is paired with all states in $R(x_0, \epsilon)$ that can be reached from x_0 via zero, one, or more unobservable transitions)
  3. From any state $q_{iobs} \in Q_{iobs}$ (recall $q_{iobs} \subseteq X \times X$) of the initial-state estimator, the next state for any $\sigma_o \in \Sigma_o$ is captured by

$$\delta_{iobs}(q_{iobs}, \sigma_o) = q_{iobs} \circ m_{\sigma_o}$$

- Initial-state estimator captures the set of possible initial states in G following a sequence of observations $\omega \in \Sigma_o^*$ via

$$\widehat{X}_0(\omega) = \{x_i \in X_0 \mid (x_i, x_c) \in \delta_{iobs}(Q_{0,iobs}, \omega)\}$$

Initial-State Opacity

Formal Definition and Verification [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]

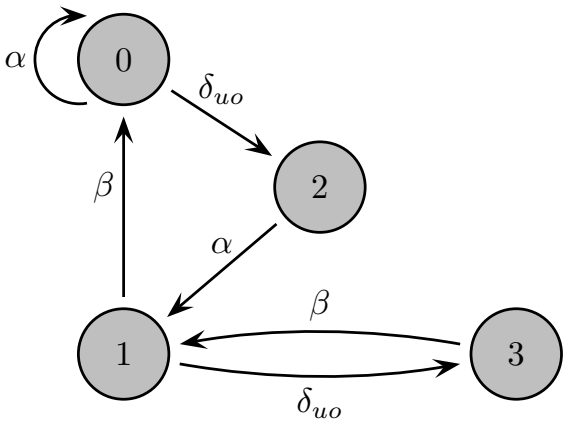
- Initial-State Opacity [Saboori and CNH, 2008]: For all $s \in L(G)$, for all $x_0 \in S_0$, it holds
$\{\delta(x_0, s) \neq \emptyset\} \Rightarrow \{\exists t \in \Sigma^*, \exists x_0' \in (X_0 \setminus S_0), \{P(t) = P(s), \delta(x_0', t) \neq \emptyset\}\}$

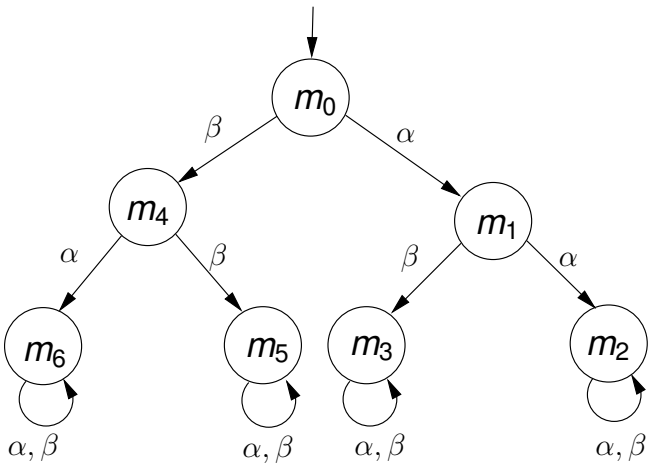
- Verification using an initial-state estimator
$G_{iobs} = (Q_{iobs}, \Sigma_o, \delta_{iobs}, Q_{0,iobs})$

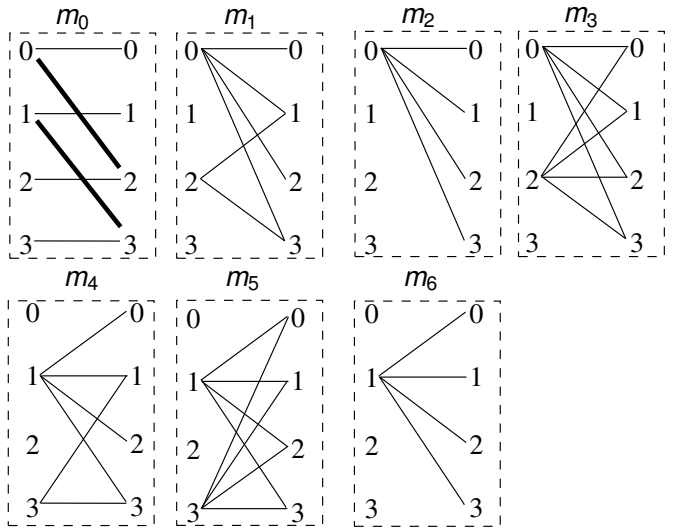
- For $q_{iobs} \in Q_{iobs}$, let $I(q_{iobs}) = \{x_i \in X_0 \mid (x_i, x_c) \in q_{iobs}\}$; G is initial-state opaque if and only if

$$\forall q_{iobs} \in Q_{iobs}, \text{ we have } I(q_{iobs}) \cap (X_0 \setminus S_0) \neq \emptyset$$

Verification of Initial-State Opacity
Example







- Not initial-state opaque wrt $S_0 = \{0\}$ (e.g., $\alpha\alpha(\alpha + \beta)^*$ reveals initial state was 0)
- Not initial-state opaque wrt $S_0 = \{1\}$ (e.g., $\beta\alpha(\alpha + \beta)^*$ reveals initial state was 1)
- Initial-state opaque wrt $S_0 = \{2\}$ or $S_0 = \{3\}$ or $S_0 = \{2, 3\}$
- Enforcement of initial-state opacity via (i) appropriate control (supervisory control) or (ii) obfuscation of observations (sensor manipulation)

Talk Outline

- Introduction and Motivation
  1. Opacity as a privacy notion
  2. Motivating examples
- Observation Models; Language-Based and State-Based Opacity
- Current-State Opacity and its Verification
  1. Current-state estimation
  2. Formal definition of current-state opacity
  3. Verification using observer (current-state estimator)
- Initial-State Opacity and its Verification
  1. Initial-state estimation
  2. Formal definition of initial-state opacity
  3. Verification using initial-state estimator
- Other State-Based Notions of Opacity
- Ongoing Research and Challenges

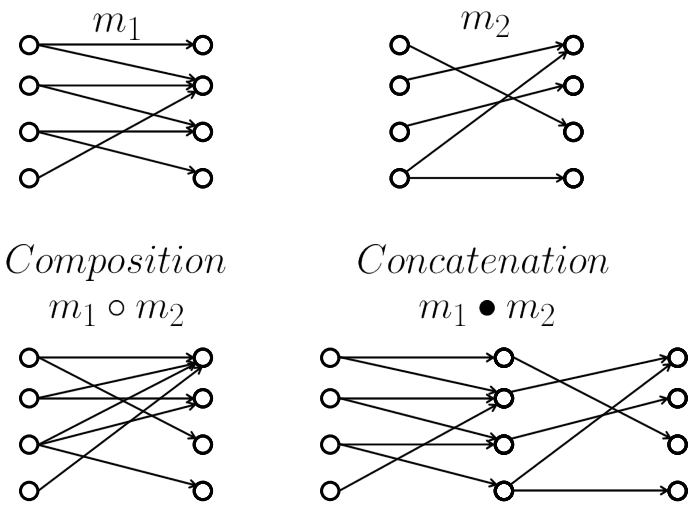
Recursive State Estimation

State Mappings, Composition, Concatenation [CNH, 2020]

- 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 $x_{i_2} \in X$)

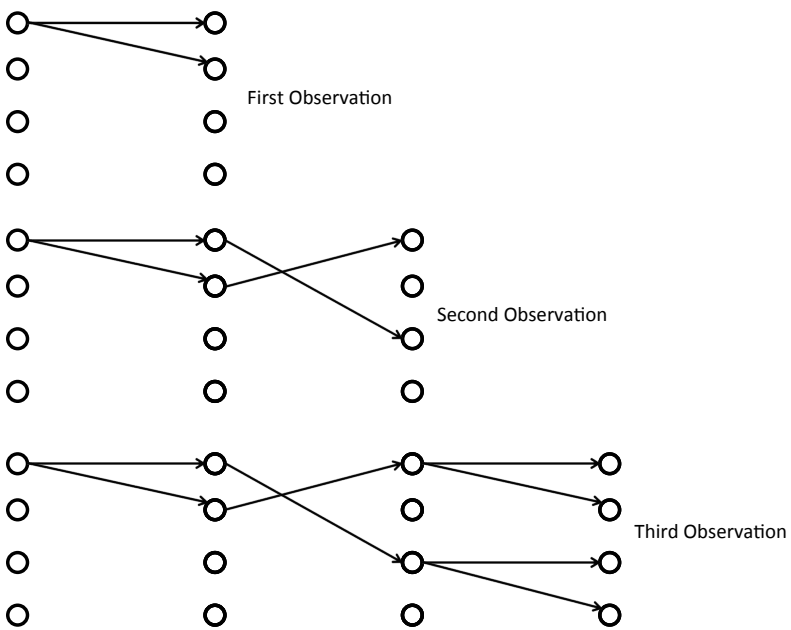


Recursive State Estimation
Output State Mappings and Induced Trellis Diagrams

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

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

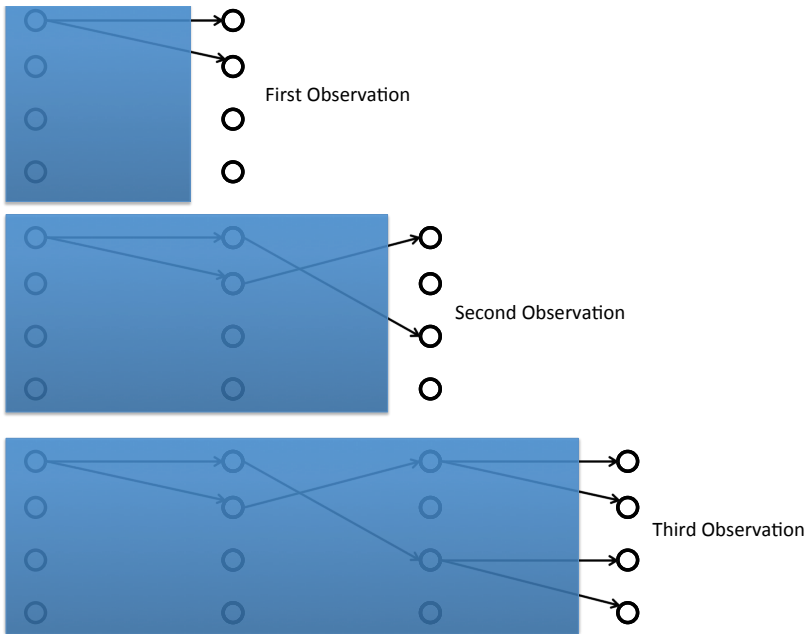
- State mappings corresponding to a sequence of observable events, $\omega = \sigma_{i_1}\sigma_{i_2}...\sigma_{i_k} \in \Sigma_o^*$ can be concatenated as $m_{\sigma_{i_1}} \bullet m_{\sigma_{i_2}} \bullet ... \bullet m_{\sigma_{i_k}}$
- Resulting construction captures matching sequences of states



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

Current State Estimation
Prunning of Trellis Diagrams

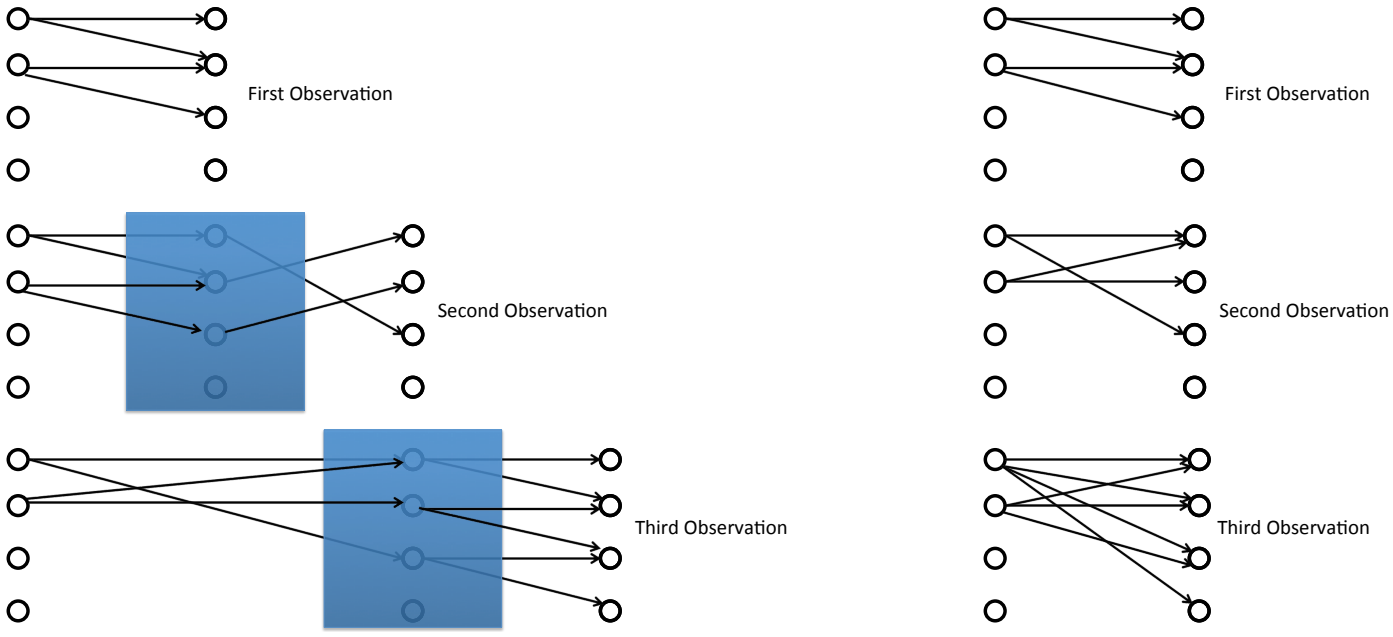
- Trellis diagrams can be pruned from parts not useful for task at hand
- Current state estimation only needs latest stage



- 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

- Introduction and Motivation
  1. Opacity as a privacy notion
  2. Motivating examples
- Observation Models; Language-Based and State-Based Opacity
- Current-State Opacity and its Verification
  1. Current-state estimation
  2. Formal definition of current-state opacity
  3. Verification using observer (current-state estimator)
- Initial-State Opacity and its Verification
  1. Initial-state estimation
  2. Formal definition of initial-state opacity
  3. Verification using initial-state estimator
- Other State-Based Notions of Opacity
- Ongoing Research and Challenges

- Extensions to other DES:
  1. Petri net models (and other purely event-driven systems)
  2. Stochastic systems (stochastic notions of opacity)
  3. Timed systems (timed notions of opacity)
- Extensions to Cyberphysical systems:
  1. Quantifying opacity (measures for opacity)
  2. Appropriate notions of opacity and their verification
- Extensions to distributed/decentralized observation settings
  1. Role of modularity and/or other system structure
  2. Resiliency to transmission delays, packet drops, errors in communication exchanges, faulty/malicious components
- Opacity Enforcement Strategies
  1. Supervisory control
  2. Obfuscation
  3. Game Formulations

- C. N. Hadjicostis, *Estimation and Inference in Discrete Event Systems*, Springer, 2020 (Chapters 4 and 8).
- J. Romain, J.-J. Lesage, and J.-M. Faure, *“Overview of discrete event systems opacity: Models, validation, and quantification,"* Annual Reviews in Control, 41: 135–146, 2016.
- S. Lafortune, F. Lin, and C. N. Hadjicostis, *“On the history of diagnosability and opacity in discrete event systems,"* Annual Reviews in Control, 45: 257–266, 2018.

Related Literature


E. Badouel, M. Bednarczyk, A. Borzyszkowski, B. Caillaud, and Ph. Darondeau, *"Concurrent secrets,"* Discrete Event Dynamic Systems, 17(4):425–446, December 2007.


J. W. Bryans, M. Koutny, and P. Y. Ryan, *"Modelling opacity using Petri nets,"* Electronic Notes in Theoretical Computer Science, 121: 101–115, 2005.


J. Bryans, M. Koutny, and P. Ryan,*" Modelling dynamic opacity using Petri nets with silent actions,"* Formal Aspects in Security and Trust, 173: 159–172, 2005.


J. Bryans, M. Koutny, L. Mazare, and P. Ryan, *"Opacity generalised to transition systems,"* International Journal of Information Security, 7(6): 421–435, November 2008.


R. Canetti, L. Cheung, D. Kaynar, M. Liskov, N. Lynch, O. Pereira, and R. Segala, *" Analyzing security protocols using time-bounded task-PIOAs,"* Discrete Event Dynamic Systems, 18(1):111–159, 2008.


C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Springer Science & Business Media, 2009.


F. Cassez, J. Mullins, and O. Roux, *"Synthesis of non-interferent distributed systems,"* Computer Network Security, 1: 159–170, August 2007.


J. Dubreil, Ph. Darondeau, and H. Marchand, *"Opacity enforcing control synthesis,"* in Proceedings of the 9th International Workshop on Discrete Event Systems (WODES), pp. 28–35, 2008.


J. Dubreil, Ph. Darondeau, and H. Marchand, *"Supervisory control for opacity,"* IEEE Transactions on Automatic Control, 55(5): 1089–1100, 2010.


N. B. Hadj-Alouane, S. Lafrance, L. Feng, J. Mullins, and M. M. Yeddes, *"On the verification of intransitive noninterference in multilevel security,"* IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 35(5):948–958, October 2005.


C. N. Hadjicostis, *Estimation and Inference in Discrete Event Systems*, Springer, 2020.

Related Literature (2)

C. Keroglou and C. N. Hadjicostis, *"Probabilistic system opacity in discrete event systems,"* Discrete Event Dynamic Systems, 28(2): 289–314, 2018.

F. Lin, *"Opacity of discrete event systems and its applications,"* Automatica, 47(3): 496–503, 2011.

F. Lin and W. M. Wonham, *"On observability of discrete-event systems,"* Information Sciences, 44(3): 173–198, 1988.

A. Saboori, *"Verification and Enforcement of State-Based Notions of Opacity in Discrete Event Systems,"* PhD Diss., University of Illinois at Urbana-Champaign, 2011.

A. Saboori and C. N. Hadjicostis, *"Notions of security and opacity in discrete event systems,"* in Proceedings of the 46th IEEE Conference on Decision and Control (CDC), pp. 5056-5061, 2007.

A. Saboori and C. N. Hadjicostis, *"Verification of initial-state opacity in security applications of DES,"* in Proceedings of the 9th International Workshop on Discrete Event Systems (WODES), pp. 328–333, 2008.

A. Saboori and C. N. Hadjicostis, *"Coverage analysis of mobile agent trajectory via state-based opacity formulations,"* Control Engineering Practice, 19(9): 967–977, 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, 2011.

A. Saboori and C. N. Hadjicostis, *"Verification of initial-state opacity in security applications of discrete event systems,"* Information Sciences, 246: 115–132, 2013.

A. Saboori and C. N. Hadjicostis, *"Current-state opacity formulations in probabilistic finite automata,"* IEEE Transactions on Automatic Control, 59(1): 120–133, 2014.

Related Literature (3)


D. Thorsley and D. Teneketzis, *"Intrusion detection in controlled discrete event systems,"* in Proceedings of the 45th IEEE Conference on Decision and Control (CDC), pp. 6047–6054, 2006.


Y. Tong, Z. Li, C. Seatzu, and A. Giua, *"Decidability of opacity verification problems in labeled Petri net systems,"* Automatica, 80: 48–53, 2017.


Y. Tong, Z. Li, C. Seatzu, and A. Giua, *"Current-state opacity enforcement in discrete event systems under incomparable observations,"* Discrete Event Dynamic Systems, 28(2): 161–182, 2018.


Y.-C. Wu and S. Lafortune, *"Comparative analysis of related notions of opacity in centralized and coordinated architectures,"* Discrete Event Dynamic Systems, 23(3): 307–339, 2013.


Y.-C. Wu and S. Lafortune, *"Synthesis of insertion functions for enforcement of opacity security properties,"* Automatica, 50(5): 1336–1348, 2014.


Y.-C. Wu, K. A. Sankararaman, and S. Lafortune, *"Ensuring privacy in location-based services: An approach based on opacity enforcement,"* IFAC Proceedings Volumes, 47(2): 33–38, 2014.


X. Yin fand S. Lafortune, *"A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems,"* IEEE Transactions on Automatic Control, 61(8): 2140–2154, 2015.


X. Yin, Z. Li, W. Wang, and S. Li, *"Infinite-step opacity and K-step opacity of stochastic discrete-event systems,"* Automatica, 99: 266–274, 2019.