Opacity and Supervisory Control, or

Enforcing confidential properties for Information Systems

Hervé Marchand

Inria Rennes - Bretagne Atlantique, France

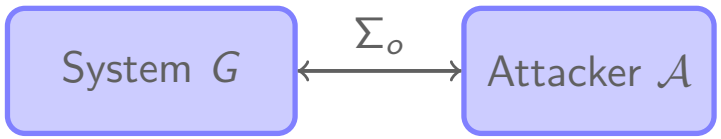
30 years of the Ramadge-Wonham Theory of Supervisory Control
IEEE CDC 2017

Security of Information Systems

- Emergence of new services
  - (E-)voting system
  - Medical card (medical information storage)
  - E-passport (biometric & person. information)
  - Web-services: E-Bancking, on-line payment, travel agency, etc
  - IOT (connected devices)
- Have to deal with critical information that should not flow, be erased or corrupted
- Security Aspects for information systems are classified into 3 categories:
  - *Integrity*: something illegal cannot be performed by a user
  - *Confidentiality*: some secret information cannot be inferred by a user
  - *Availability*: a user can always perform the actions that are allowed by the other security policies

Information flow

- Context: Given a system $\mathcal{G}$ and an attacker $\mathcal{A}$ who
  - knows the model of $\mathcal{G}$ and
  - observes $\mathcal{G}$ and interacts with $\mathcal{G}$ through Σ_o



- Confidentiality information: secret properties
  - State configurations (values of secret variables)
  - Occurrence of an event (e.g. password file is unlocked),
  - Trajectories, sequences of the system

- Information flow: an attacker is able to infer confidential information based on observations and the knowledge of the system.

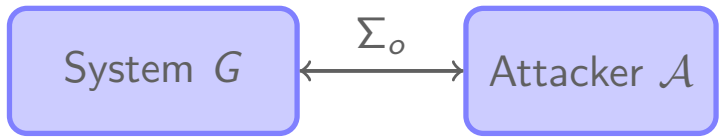
$\Rightarrow$ From non-interference prop. [GM82] to opacity prop. [BKMR08]

[GM82] J. Goguen, J. Meseguer. Security policies and security models. Proc of IEEE Symp. on Security and Privacy, 1982.

[BKMR08]J. Bryans, M. Koutny, L. Mazaré, and P. Ryan. Opacity generalised to transition systems. Int. Journal of Information Security, 7(6):421-435, 2008

Information flow

- Context: Given a system $\mathcal{G}$ and an attacker $\mathcal{A}$ who
  - knows the model of $\mathcal{G}$ and
  - observes $\mathcal{G}$ and interacts with $\mathcal{G}$ through Σ_o



- Confidentiality information: secret properties
  - State configurations (values of secret variables)
  - Occurrence of an event (e.g. password file is unlocked),
  - Trajectories, sequences of the system

- Information flow: an attacker is able to infer confidential information based on observations and the knowledge of the system.

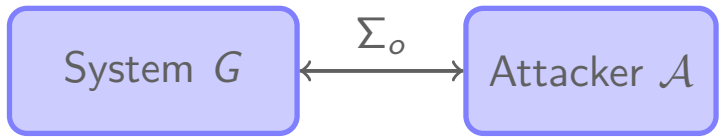
$\Rightarrow$ From non-interference prop. [GM82] to opacity prop. [BKMR08]

[GM82] J. Goguen, J. Meseguer. Security policies and security models. Proc of IEEE Symp. on Security and Privacy, 1982.

[BKMR08]J. Bryans, M. Koutny, L. Mazaré, and P. Ryan. Opacity generalised to transition systems. Int. Journal of Information Security, 7(6):421-435, 2008

Information flow

- Context: Given a system $\mathcal{G}$ and an attacker $\mathcal{A}$ who
  - knows the model of $\mathcal{G}$ and
  - observes $\mathcal{G}$ and interacts with $\mathcal{G}$ through Σ_o



- Confidentiality information: secret properties
  - State configurations (values of secret variables)
  - Occurrence of an event (e.g. password file is unlocked),
  - Trajectories, sequences of the system

- Information flow: an attacker is able to infer confidential information based on observations and the knowledge of the system.

$\Rightarrow$ From non-interference prop. [GM82] to opacity prop. [BKMR08]

[GM82] J. Goguen, J. Meseguer. Security policies and security models. Proc of IEEE Symp. on Security and Privacy, 1982.

[BKMR08]J. Bryans, M. Koutny, L. Mazaré, and P. Ryan. Opacity generalised to transition systems. Int. Journal of Information Security, 7(6):421-435, 2008

Outline

- Introduction
- Opacity Problem
- Supervisory Control for opacity
- Ensuring opacity by Dynamic filtering
- Conclusion & Perspectives

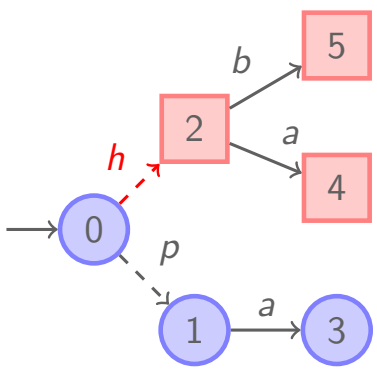
State-Based (Language-Based) Opacity

Definition (State-Based opacity)

The secret $S \subseteq Q$ (or $\mathcal{L}_S(\mathcal{G})$) is opaque w.r.t. $\mathcal{G}$ and Σ_o if [BKMR08]

$$\forall t \in \mathcal{L}_S(\mathcal{G}), \exists s \in \mathcal{L}(\mathcal{G}) : \Pi_{\Sigma_o}(s) = \Pi_{\Sigma_o}(t) \wedge s \notin \mathcal{L}_S(\mathcal{G}).$$

Example



$$\Sigma = \{h, p, a, b\},\ \Sigma_o = \{a, b\}$$

$$\mathcal{L}_S(\mathcal{G}) = \Sigma^*.h.\Sigma^*$$

[BKMR08]J. Bryans, M. Koutny, L. Mazaré, and P. Ryan. Opacity generalised to transition systems. International Journal of Information Security, 7(6):421-435, 2008

Other notions of Opacity

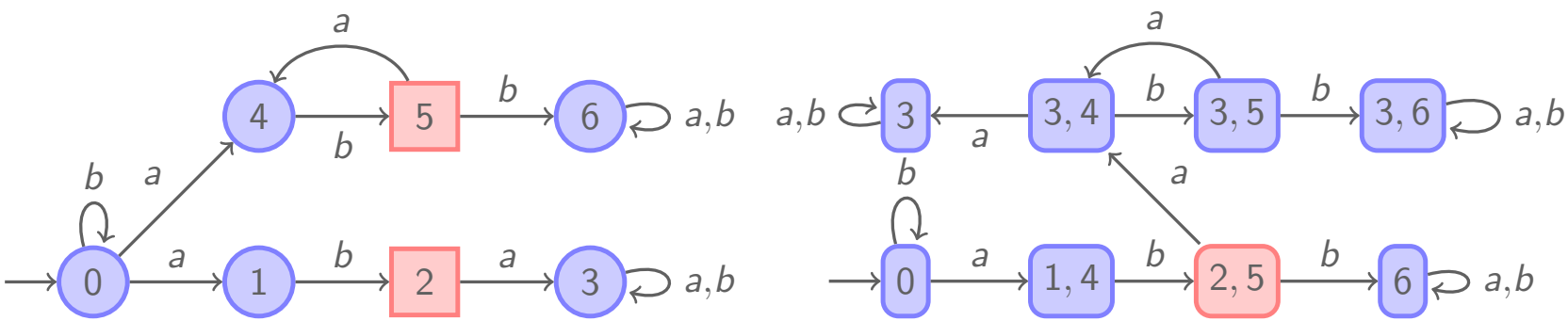
- K-Step opacity [SH07]
  $\Rightarrow$ the system traversed the secret less that K observations ago
- Infinite step opacity [SH07]
  $\Rightarrow$ the system traversed the secret in the past
- Initial opacity [SH13]
  $\Rightarrow$ similar to infinite opacity but the secret is a subset of the initial states.
- Initial & Final opacity [WL12]
- Secrecy, Non-Blocking opacity, etc [JLF16]

[SH07] A. Saboori, C. Hadjicostis. Notions of security and opacity in discrete event systems. In: CDC'07: 46th IEEE Conf. Decision and Control, pp 5056-5061, 2007
[SH09] A. Saboori, C. Hadjicostis Verification of infinite-step opacity and analysis of its complexity. In: Dependable Control of Discrete Systems, 2009
[SH13] A. Saboori and C. Hadjicostis. Verification of initial-state opacity in security applications of discrete event systems. Inf. Sci., 246:115-132, 2013.
[WL12]Y. Wu, S Lafortune. Comparative analysis of related notions of opacity in centralized and coordinated architectures DEDS 23-3, pp 307-339. 2012
[JLF16] R. Jacob, J.-J. Lesage, and J.-M. Faure. Overview of Discrete Event Systems Opacity: models, validation, and quantification. Annual Reviews in Control, April 2016.

Verification of (State-Based) Opacity

Algorithm

1. Determinization of $\mathcal{G}$
2. check whether a macro-state $F \subseteq S$ is reachable



Theorem

Checking opacity is PSPACE complete [ATVA'09]

$\Rightarrow$ Reduction from universality problem

[ATVA'09] F. Cassez, J. Dubreil, H. Marchand. Dynamic Observers for the Synthesis of Opaque Systems. In 7th International Symposium on Automated Technology for Verification and Analysis, LNCS, Vol 5799, pp. 352-367, 2009.

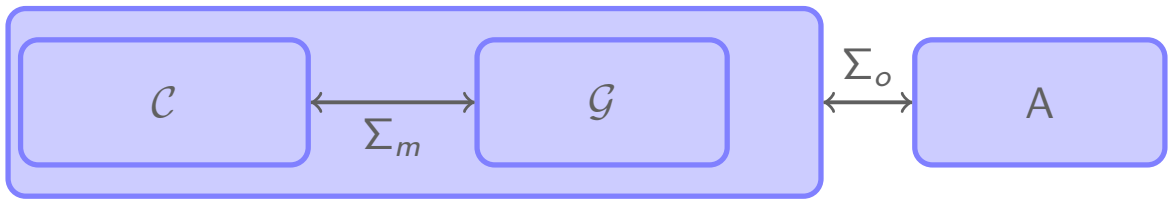
Outline

- Introduction
- Opacity Problem
- Supervisory Control for opacity
- Ensuring opacity by Dynamic filtering
- Conclusion & Perspectives

Supervisory Control for opacity

[Wodes'08], [TAC'10]

Compute an Access Control using Supervisory Control Theory



Control problem

Hyp: $\mathcal{C}$ observes $\Sigma_m \subseteq \Sigma$ and controls $\Sigma_c \subseteq \Sigma_m$,

Pb: Compute a maximally permissive access control C observing Σ_m and controlling $\Sigma_c \subseteq \Sigma_m$ s.t. S is opaque w.r.t. $\mathcal{G} \times \mathcal{C}$ and Σ_o.

Maximality: $\forall \mathcal{C}'$, S is opaque for $\mathcal{L}(\mathcal{G} \times \mathcal{C}')$ $\mathcal{L}(\mathcal{G} \times \mathcal{C}') \subseteq \mathcal{L}(\mathcal{G} \times \mathcal{C})$

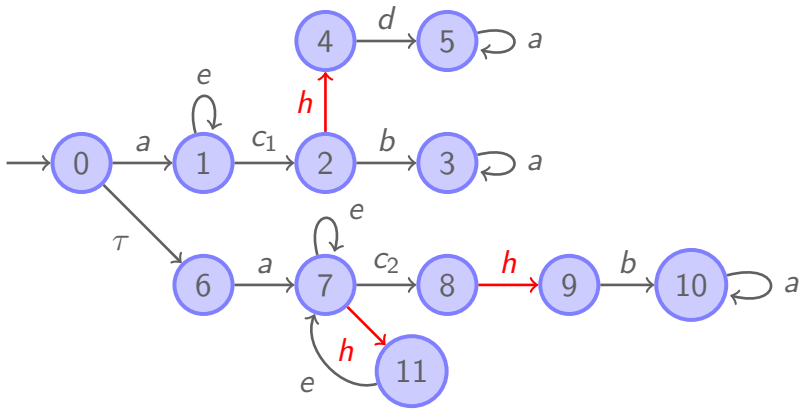
[Wodes'08] Opacity Enforcing Control Synthesis. In Workshop on Discrete Event Systems, WODES'08, pp. 28-35, Sweden, 2008
[TAC'10] J. Dubreil, P. Darondeau, H. Marchand. Supervisory Control for Opacity. IEEE Transactions on Automatic Control, 55(5):1089-1100, May 2010

Supervisory Control for Opacity (Intuition)

Problem

Compute a maximally permissive access control $\mathcal{C}$ observing Σ_m and controlling $\Sigma_c \subseteq \Sigma_m$ s.t. S is opaque w.r.t. $\mathcal{G} \times \mathcal{C}$ and Σ_o.

Example



$$\Sigma_o = \{a, b, d, e\}$$
$$\Sigma_m = \{a, c_1, c_2, b, d, e\}$$
$$\Sigma_c = \{b, c_1, c_2, e\}$$

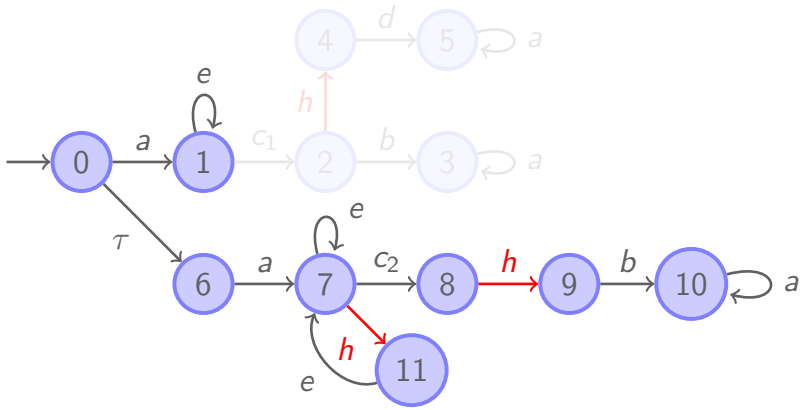
Secret: $\mathcal{L}_S(\mathcal{G}) = \Sigma^*.h.\Sigma^*$

Supervisory Control for Opacity (Intuition)

Problem

Compute a maximally permissive access control $\mathcal{C}$ observing Σ_m and controlling $\Sigma_c \subseteq \Sigma_m$ s.t. S is opaque w.r.t. $\mathcal{G} \times \mathcal{C}$ and Σ_o.

Example



$$\Sigma_o = \{a, b, d, e\}$$
$$\Sigma_m = \{a, c_1, c_2, b, d, e\}$$
$$\Sigma_c = \{b, c_1, c_2, e\}$$

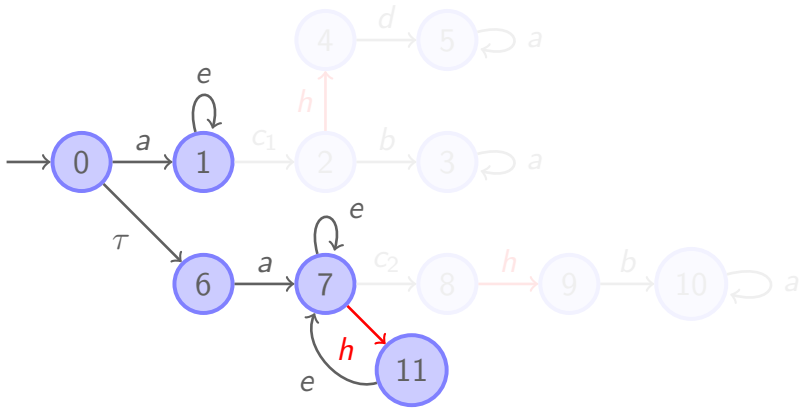
Secret: $\mathcal{L}_S(\mathcal{G}) = \Sigma^*.h.\Sigma^*$

Supervisory Control for Opacity (Intuition)

Problem

Compute a maximally permissive access control $\mathcal{C}$ observing Σ_m and controlling $\Sigma_c \subseteq \Sigma_m$ s.t. S is opaque w.r.t. $\mathcal{G} \times \mathcal{C}$ and Σ_o.

Example



$$\Sigma_o = \{a, b, d, e\}$$
$$\Sigma_m = \{a, c_1, c_2, b, d, e\}$$
$$\Sigma_c = \{b, c_1, c_2, e\}$$

Secret: $\mathcal{L}_S(\mathcal{G}) = \Sigma^*.h.\Sigma^*$

Existence of a Supremal Sub-language

Property

There exists a supremal prefix-closed, opaque, observable and controllable sub-language of $\mathcal{L}(\mathcal{G})$ denoted

$$\text{CO-OP}^{\uparrow}(\mathcal{L}(\mathcal{G}), \mathcal{L}_S(\mathcal{G}), \Sigma_o, \Sigma_m, \Sigma_c) \subseteq \Sigma^*$$

Opacity is stable under arbitrary union, like observability (normality case) and controllability

Is this language regular? Effective computation?

- $CO^{\uparrow}(\mathcal{L}(\mathcal{G}), \Sigma_m, \Sigma_c)(H)$ supremal prefix-closed observable and controllable sub-language of $\mathcal{L}(\mathcal{G})$ included in H

- $Op^{\uparrow}(\mathcal{L}_S(\mathcal{G}), \Sigma_o)(H)$ supremal opaque sublanguage of H

$$\mathcal{L}(\mathcal{G}) \cap P_{\Sigma_o}^{-1}(\mathcal{L}_{2^S}(Det_{\Sigma_o}(\mathcal{G}))).\Sigma^*$$

- The operator $CO^{\uparrow}(\mathcal{L}(\mathcal{G}), \Sigma_m, \Sigma_c) \circ Op^{\uparrow}(\mathcal{L}_S(\mathcal{G}), \Sigma_o)(H)$ is monotonic contractive on prefix-closed sub-languages of $\mathcal{L}(\mathcal{G})$.

$\Rightarrow$ This operator has a greatest fix point $K(\mathcal{L}(\mathcal{G}), \mathcal{L}_S(\mathcal{G}))$

Theorem

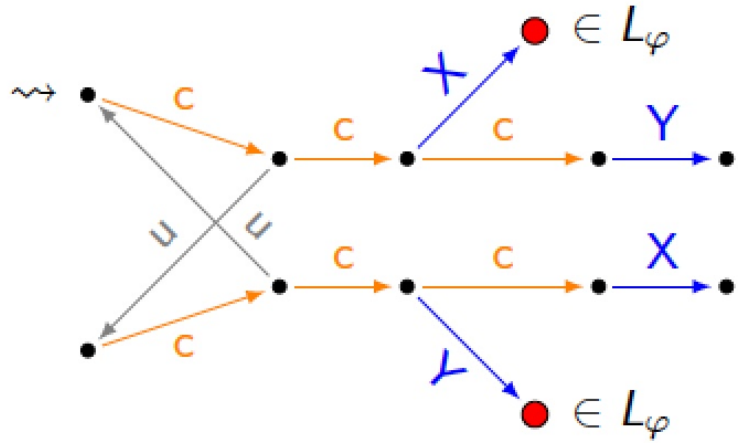
$$K(\mathcal{L}(\mathcal{G}), \mathcal{L}_S(\mathcal{G})) = \text{CO-OP}^{\uparrow}(\mathcal{L}(\mathcal{G}), \mathcal{L}_S(\mathcal{G}), \Sigma_o, \Sigma_m, \Sigma_c)$$

The greatest fix-point computation starting from $\mathcal{L}(G)$ may not terminate

Counter Example

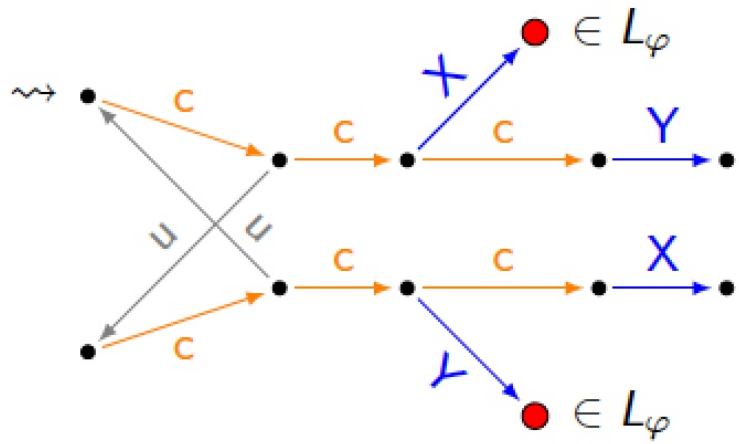
Hypothesis:

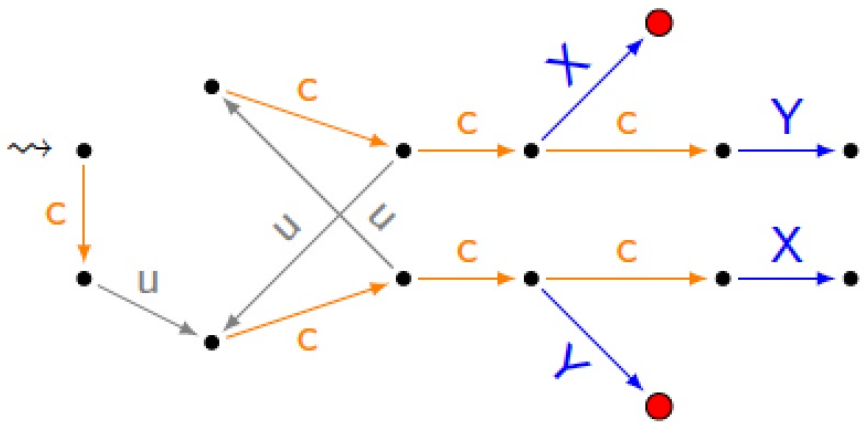
- $\Sigma_m = \Sigma$
- $\Sigma_a = \{c, X, Y\}$
- $\Sigma_c = \{c\}$



Counter Example

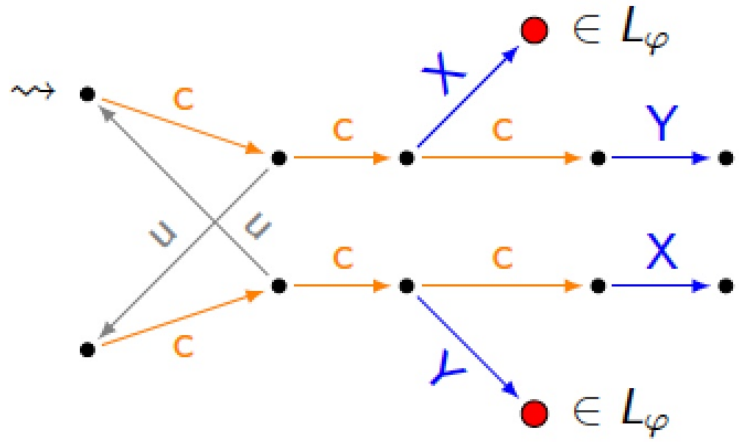
after one iteration

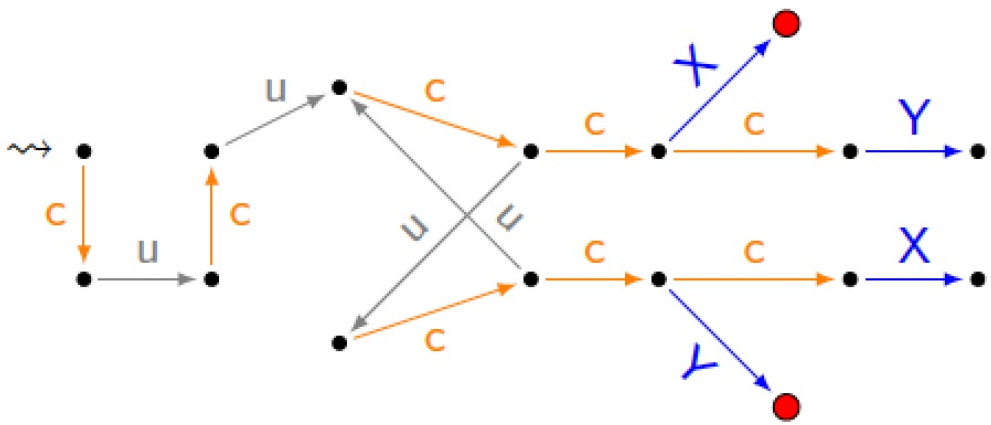




Counter Example

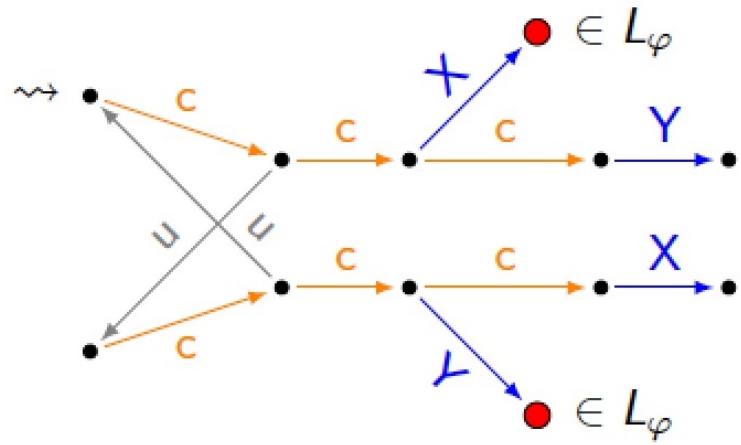
after two iterations

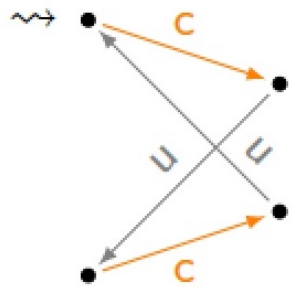




Counter Example

not two consecutive c transitions





Results

The greatest fix-point computation terminates in the cases:

- If $\Sigma_c \subseteq \Sigma_m \subseteq \Sigma_o \subseteq \Sigma$ then

$$CO^{\uparrow}(\mathcal{L}(\mathcal{G}), \Sigma_m, \Sigma_c) \circ Op^{\uparrow}(\mathcal{L}_S(\mathcal{G}), \Sigma_o)(\mathcal{L}(G)) = K(\mathcal{L}(\mathcal{G}), \mathcal{L}_S(\mathcal{G}))$$

- If $\Sigma_o \subseteq \Sigma_c \subseteq \Sigma_m \subseteq \Sigma$ then

$$Op^{\uparrow}(\mathcal{L}_S(\mathcal{G}), \Sigma_o)(\mathcal{L}(\mathcal{G}) = K(\mathcal{L}(\mathcal{G}), \mathcal{L}_S(\mathcal{G}))$$

- more general result

Theorem

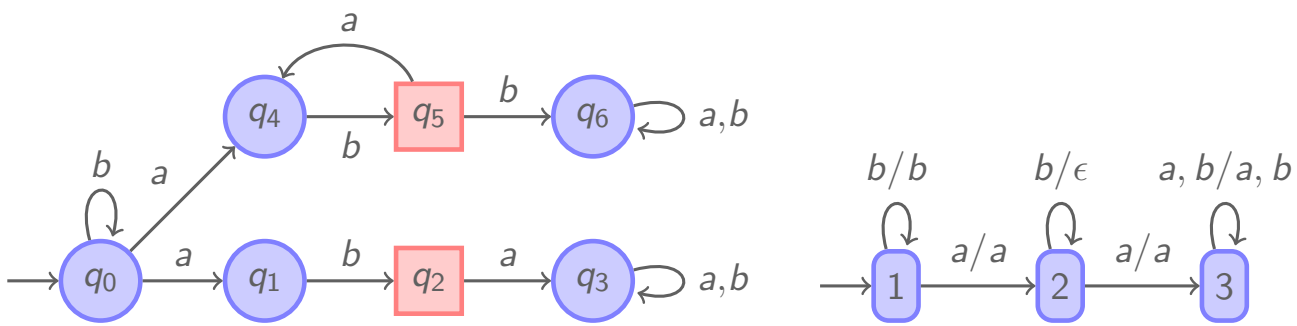
If $\Sigma_c \subseteq \Sigma_m$ and $\Sigma_m \subseteq \Sigma_o$, or $\Sigma_o \subseteq \Sigma_m$, there exists a maximal controller $\mathcal{C}$ s.t. S is opaque w.r.t. $\mathcal{G} \times \mathcal{C}$ and Σ_o

Outline

- Introduction
- Opacity Problem
- Supervisory Control for opacity
- Ensuring opacity by Dynamic filtering
- Conclusion & Perspectives

Ensuring opacity by Dynamic filtering

[ATVA'09], [FMSD'12]



- Static Filter: $\Sigma_o = \{a\}$ or $\Sigma_o = \{b\} \Rightarrow S$ is opaque
- Dynamic Filter: Hide"b" after the observation of an "a" and keep everything observable after the observation of an "a"

Problem

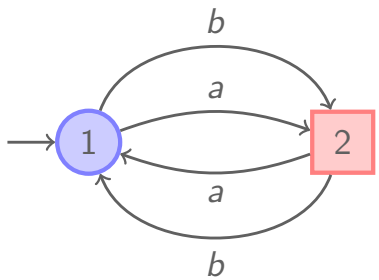
Computing a Dynamic filter so that S is opaque w.r.t. $G \times D$ and Σ_o

[TC08] F. Cassez and S. Tripakis. Fault diagnosis with static and dynamic diagnosers. Fundamenta Informaticae, 88(4):497-540, November 2008.

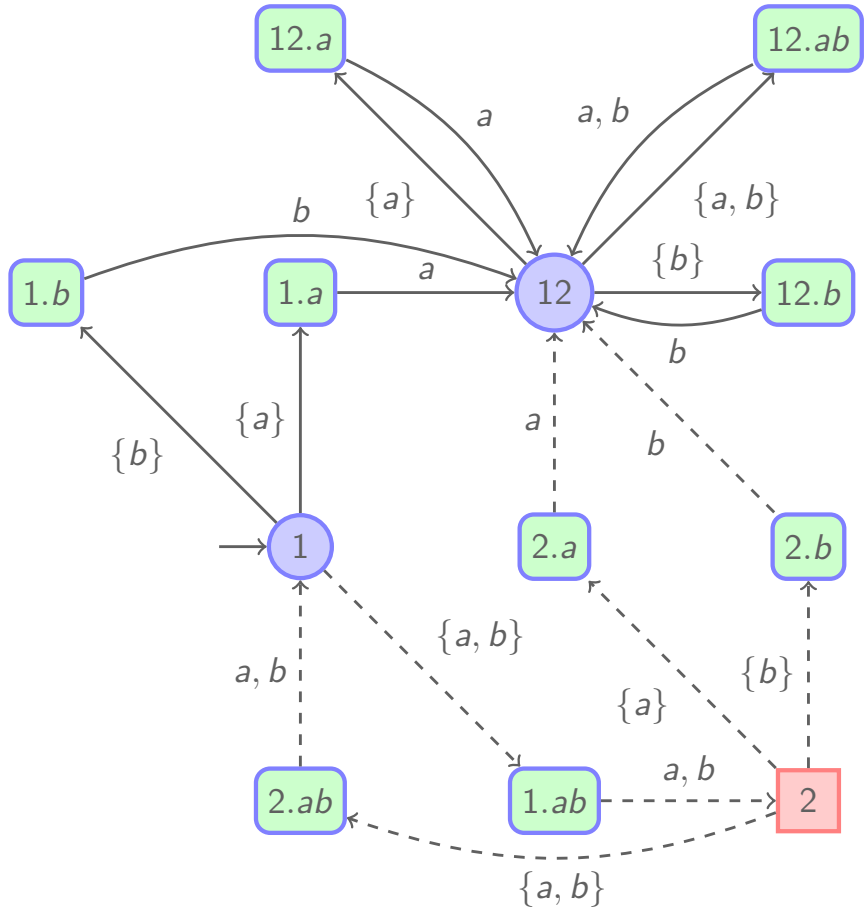
[FMSD12] F. Cassez, J. Dubreil, and H. Marchand. Synthesis of opaque systems with static and dynamic masks. Formal Methods in System Design, 40(1), 2012

Ensuring opacity by Dynamic filtering

- Reduction to a safety 2-player game



(a): the LTS $\mathcal{G}$

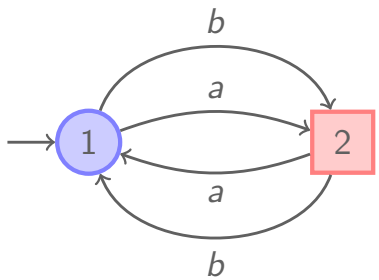


(b): The game LTS

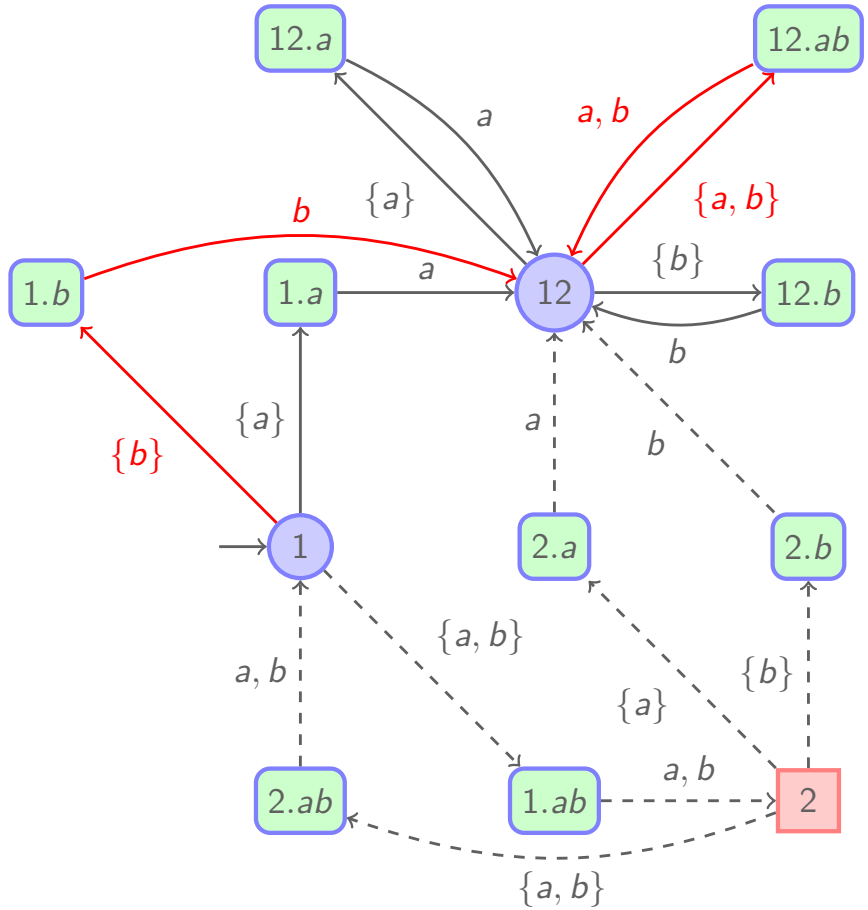
⇒ Solving the Dynamic filter problem is EXPTIME

Ensuring opacity by Dynamic filtering

- Reduction to a safety 2-player game



(a): the LTS $\mathcal{G}$



(b): The game LTS

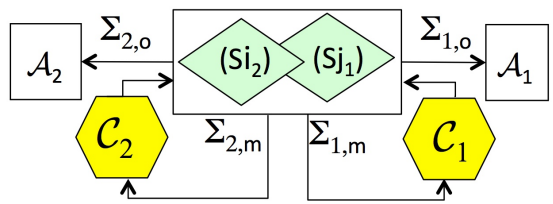
⇒ Solving the Dynamic filter problem is EXPTIME

Outline

- Introduction
- Opacity Problem
- Supervisory Control for opacity
- Ensuring opacity by Dynamic filtering
- Conclusion & Perspectives

Conclusion & Perspectives

- Summary
  - Ensuring Opacity by control [Wodes'08], [TAC'10]
  - Ensuring Opacity Dynamic Filtering [Amast08], [FMSD12]
  - Insertion Function [WL12]
- Perspectives
  - Opacity control problem
    - Remove the assumption $\Sigma_m \subseteq \Sigma_o$, or $\Sigma_o \subseteq \Sigma_m$ [TMLG+16]
    - Distributed control of Concurrent secrets



preliminary work in [BBB+07]

  - Active Attacker
    - Ability to change the observable status of events [KWKL16]
    - Forcing event
    - More powerful observation aspects
  - Opacity of systems handling data

[WL12] Y. Wu and S. Lafortune. Enforcement of opacity properties using insertion functions. In 51st IEEE Conf. on Decision and Contr., pages 6722-6728, 2012.

[TMLG+16] Y. Tong, Z. Ma, Z. Li, C. Seatzu, and A. Giua. Supervisory enforcement of current-state opacity with uncomparable observations. In Proc of 13th Workshop on Discrete Event Systems, WODES'16, 2016.

[BBB+07] E. Badouel, M. Bednarczyk, A. Borzyszkowski, B. Caillaud, and P. Darondeau. Concurrent secrets. Discrete Event Dynamic Systems, 17(4):425-446, December 2007.

[KWKL16] C. Kawakami, Y.Wu, R. Kwong, and S. Lafortune. Detection and prevention of actuator enablement attacks in supervisory control systems. In Proc of 13th Workshop on Discrete Event Systems, WODES'16, 2016.