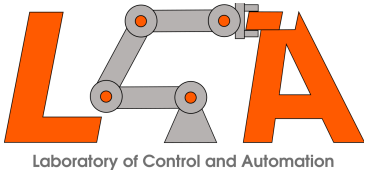
Robust Failure Diagnosis of Discrete Event Systems and Its Applications

João Carlos Basilio

Workshop on Analysis and Control for Resilience of Discrete Event Systems



Universidade Federal
do Rio de Janeiro

Talk outline

1. Fault Diagnosis and Robustness
2. Fault Diagnosis of Discrete-Event Systems
3. Online Diagnosis
4. Robust Fault Diagnosis of Discrete-Event Systems. Why?
5. Robust Diagnosis against intermittent loss of observation
6. Diagnosability of Networked DES
7. Conclusion

I. Fault Diagnosis and Robustness

What is a fault?

- Faut is an unexpected change of system function, usually due to physical failure or breakdown.
  - Ex: a valve gets stuck (open/closed); a sensor stops working; communication channel malfunction;

What is a fault?

- Faut is an unexpected change of system function, usually due to physical failure or breakdown.
  - Ex: a valve gets stuck (open/closed); a sensor stops working; communication channel malfunction;
- A fault hampers or disturbs the normal operation of an automatically controlled systems
- A fault either causes an unacceptable deterioration of system performance or leads to dangerous situations

What is a fault?

- Faut is an unexpected change of system function, usually due to physical failure or breakdown.
  - Ex: a valve gets stuck (open/closed); a sensor stops working; communication channel malfunction;
- A fault hampers or disturbs the normal operation of an automatically controlled systems
- A fault either causes an unacceptable deterioration of system performance or leads to dangerous situations
- Fault *vs.* Failure:
  - Fault: malfunction that may be tolerated for some time.
  - Failure: complete breakdown of a system component.

What is a fault?

- Faut is an unexpected change of system function, usually due to physical failure or breakdown.
  - Ex: a valve gets stuck (open/closed); a sensor stops working; communication channel malfunction;
- A fault hampers or disturbs the normal operation of an automatically controlled systems
- A fault either causes an unacceptable deterioration of system performance or leads to dangerous situations
- Fault *vs.* Failure:
  - Fault: malfunction that may be tolerated for some time.
  - Failure: complete breakdown of a system component.

Fault occurrences must be diagnosed ASAP

Fault Diagnosis vs. Fault Diagnosability

- Fault Diagnosis: process of detecting the occurrence of a fault

Fault Diagnosis vs. Fault Diagnosability

- Fault Diagnosis: process of detecting the occurrence of a fault
  - Fault diagnosis system: a monitoring system used to detect faults and diagnose their location $\Rightarrow$ Diagnoser

Fault Diagnosis vs. Fault Diagnosability

- Fault Diagnosis: process of detecting the occurrence of a fault
  - Fault diagnosis system: a monitoring system used to detect faults and diagnose their location $\Rightarrow$ Diagnoser
  - It is performed online

Fault Diagnosis vs. Fault Diagnosability

- Fault Diagnosis: process of detecting the occurrence of a fault
  - Fault diagnosis system: a monitoring system used to detect faults and diagnose their location $\Rightarrow$ Diagnoser
  - It is performed online
- Fault Diagnosability: is a system property that ensures that the fault can be diagnosed
  - It is performed offline
  - It requires the knowledge of a model of the system

Robustness

- It has been introduced by Zames and Francis in the 1980's

  G. Zames and B. Francis, "Feedback, minimax sensitivity, and optimal robustness," *IEEE Transactions on Automatic Control*, vol. 28, no. 5, 585–601, 1983

Robustness

- It has been introduced by Zames and Francis in the 1980's

  G. Zames and B. Francis, "Feedback, minimax sensitivity, and optimal robustness," *IEEE Transactions on Automatic Control*, vol. 28, no. 5, 585–601, 1983

  - It is associated with the ability of a system to perform without exact knowledge of plant model

Robustness

- It has been introduced by Zames and Francis in the 1980's

  G. Zames and B. Francis, "Feedback, minimax sensitivity, and optimal robustness," *IEEE Transactions on Automatic Control*, vol. 28, no. 5, 585–601, 1983

  - It is associated with the ability of a system to perform without exact knowledge of plant model
  - H_∞ methods were employed to deal with both the effects of external signal (noise/disturbances) and parameter sensitivity attenuation

Robustness

- It has been introduced by Zames and Francis in the 1980's

  G. Zames and B. Francis, "Feedback, minimax sensitivity, and optimal robustness," *IEEE Transactions on Automatic Control*, vol. 28, no. 5, 585–601, 1983

  - It is associated with the ability of a system to perform without exact knowledge of plant model
  - H_∞ methods were employed to deal with both the effects of external signal (noise/disturbances) and parameter sensitivity attenuation

- Model: $G = G_o + \Delta_G = \left(1 + M_G\right) G_o$

Robustness

- It has been introduced by Zames and Francis in the 1980's

  G. Zames and B. Francis, "Feedback, minimax sensitivity, and optimal robustness," *IEEE Transactions on Automatic Control*, vol. 28, no. 5, 585–601, 1983

  - It is associated with the ability of a system to perform without exact knowledge of plant model
  - H_∞ methods were employed to deal with both the effects of external signal (noise/disturbances) and parameter sensitivity attenuation

- Model: $G = G_o + \Delta_G = (1 + M_G)\, G_o$

How to bring robustness to Fault Diagnosis of Discrete-Event Systems?

II. Fault Diagnosis of Discrete-Event Systems

Discrete-Event Systems

- DES: Is a dynamical system whose evolution is determined by the asynchronous occurrence of events.
  - Ex: Manufacturing cell composed of conveyor belt, a processing machine and a robot arm

Discrete-Event Systems

- DES: Is a dynamical system whose evolution is determined by the asynchronous occurrence of events.
  - Ex: Manufacturing cell composed of conveyor belt, a processing machine and a robot arm
- DES are event dependent $\Rightarrow$ Better described with languages

Discrete-Event Systems

- DES: Is a dynamical system whose evolution is determined by the asynchronous occurrence of events.
  - Ex: Manufacturing cell composed of conveyor belt, a processing machine and a robot arm
- DES are event dependent $\Rightarrow$ Better described with languages
- Automaton is one of the modeling formalisms

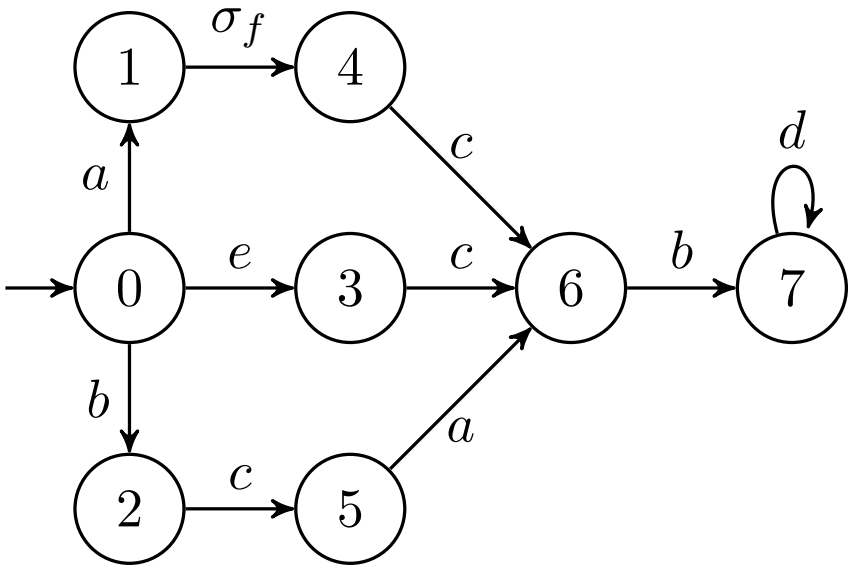
$$G = (X, \Sigma, \delta, x_0)$$

  - Kleene-closure of Σ: Σ^*
  - Language generated by G:
    $L(G) = \{s \in \Sigma^* : (\exists x \in X)[\delta(x_0, s) = x]\}$

Language diagnosability

$$G = (X, \Sigma, \delta, x_0)$$

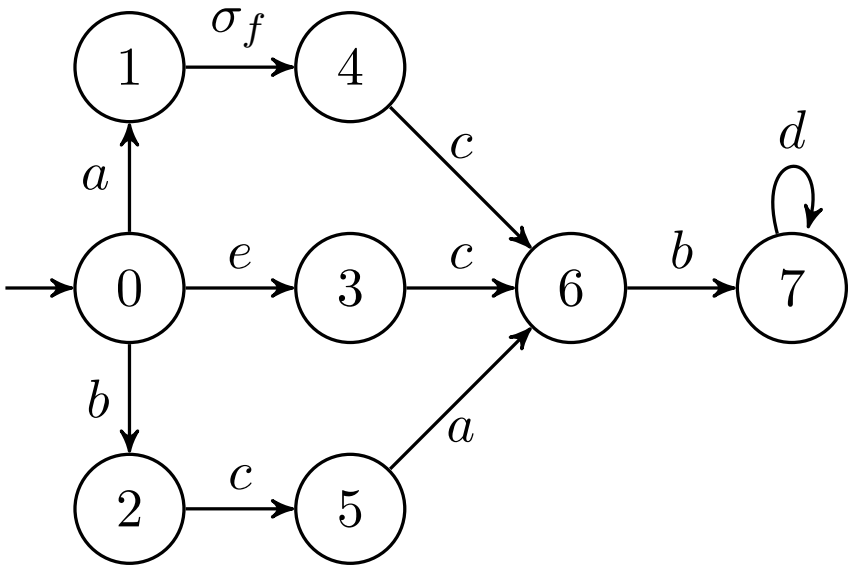
- $X = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{a, b, c, d, \sigma_f\}$
- $x_0 = 0$



Language diagnosability

$$G = (X, \Sigma, \delta, x_0)$$

- $X = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{a, b, c, d, \sigma_f\}$
- $x_0 = 0$

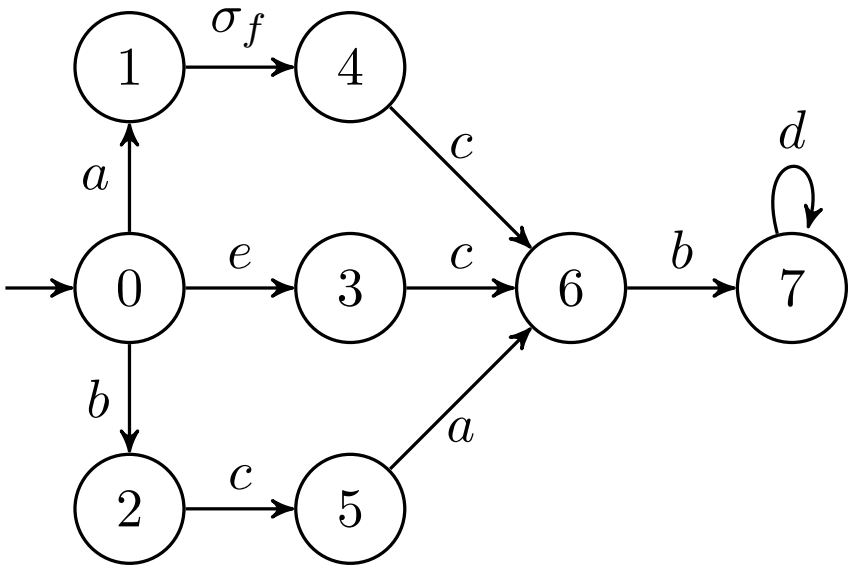


- Observable/unobservable event set partition: $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$

Language diagnosability

$$G = (X, \Sigma, \delta, x_0)$$

- $X = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{a, b, c, d, \sigma_f\}$
- $x_0 = 0$

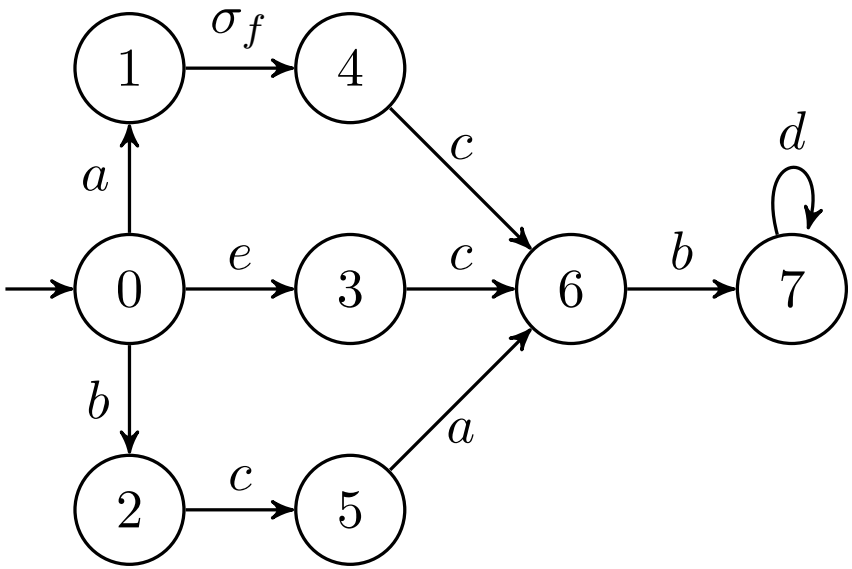


- Observable/unobservable event set partition: $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$
  - $\Sigma_o = \{c, d, e\}$; $\Sigma_{uo} = \{a, b, \sigma_f\}$
  - $\Sigma_f = \{\sigma_f\}$

Language diagnosability

$$G = (X, \Sigma, \delta, x_0)$$

- $X = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{a, b, c, d, \sigma_f\}$
- $x_0 = 0$

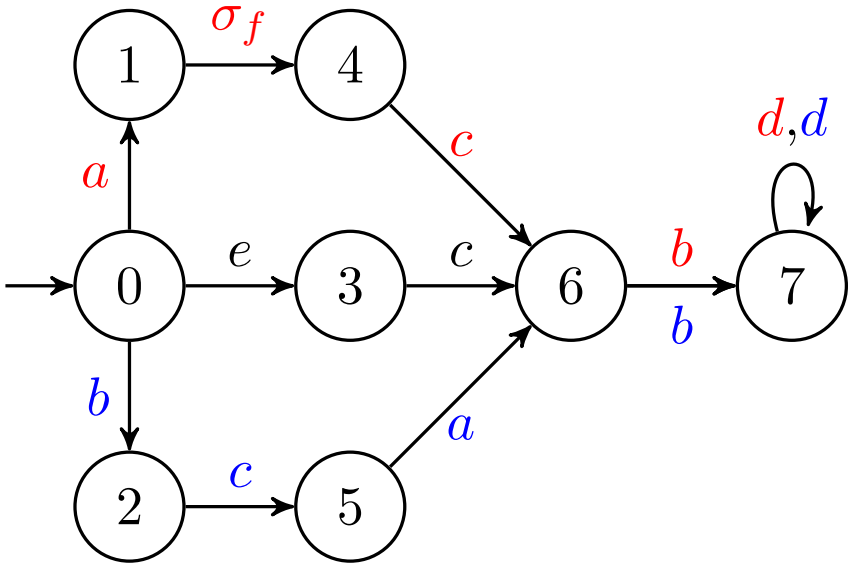


- Observable/unobservable event set partition: $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$
  - $\Sigma_o = \{c, d, e\}$; $\Sigma_{uo} = \{a, b, \sigma_f\}$
  - $\Sigma_f = \{\sigma_f\}$
  - Projection: $P_{\Sigma,\Sigma_o} : \Sigma^* \to \Sigma_o^*$

Language diagnosability

$$G = (X, \Sigma, \delta, x_0)$$

- $X = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{a, b, c, d, \sigma_f\}$
- $x_0 = 0$



- Observable/unobservable event set partition: $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$
  - $\Sigma_o = \{c, d, e\}$; $\Sigma_{uo} = \{a, b, \sigma_f\}$
  - $\Sigma_f = \{\sigma_f\}$
  - Projection: $P_{\Sigma,\Sigma_o} : \Sigma^* \to \Sigma_o^*$
  - $s_Y = a\sigma_f cbd^n$, $s_N = bcabd^n$

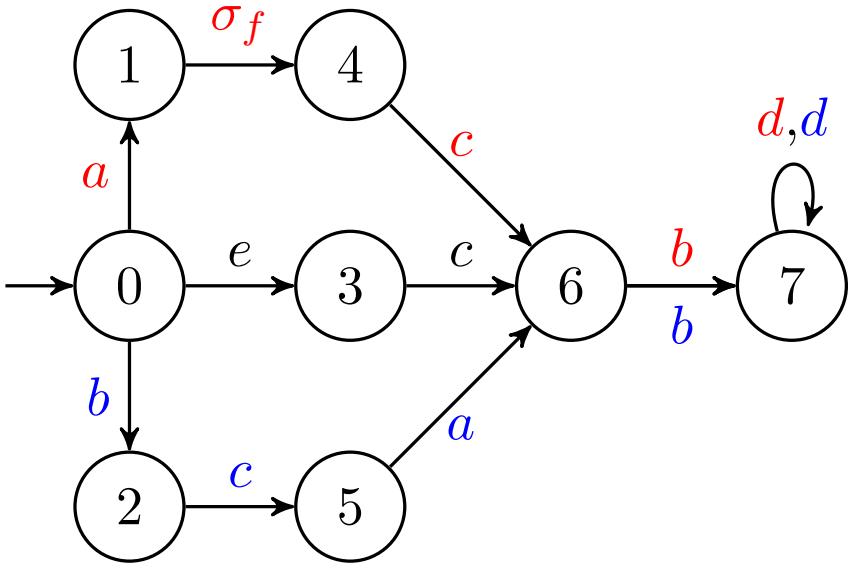
$$\Downarrow$$

$$P_{\Sigma,\Sigma_o}(s_Y) = P_{\Sigma,\Sigma_o}(s_N) = cd^n$$

Language diagnosability

$$G = (X, \Sigma, \delta, x_0)$$

- $X = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{a, b, c, d, \sigma_f\}$
- $x_0 = 0$



- Observable/unobservable event set partition: $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$
  - $\Sigma_o = \{c, d, e\}$; $\Sigma_{uo} = \{a, b, \sigma_f\}$
  - $\Sigma_f = \{\sigma_f\}$
  - Projection: $P_{\Sigma,\Sigma_o} : \Sigma^* \to \Sigma_o^*$
  - $s_Y = a\sigma_f cbd^n$, $s_N = bcabd^n$

$$\Downarrow$$

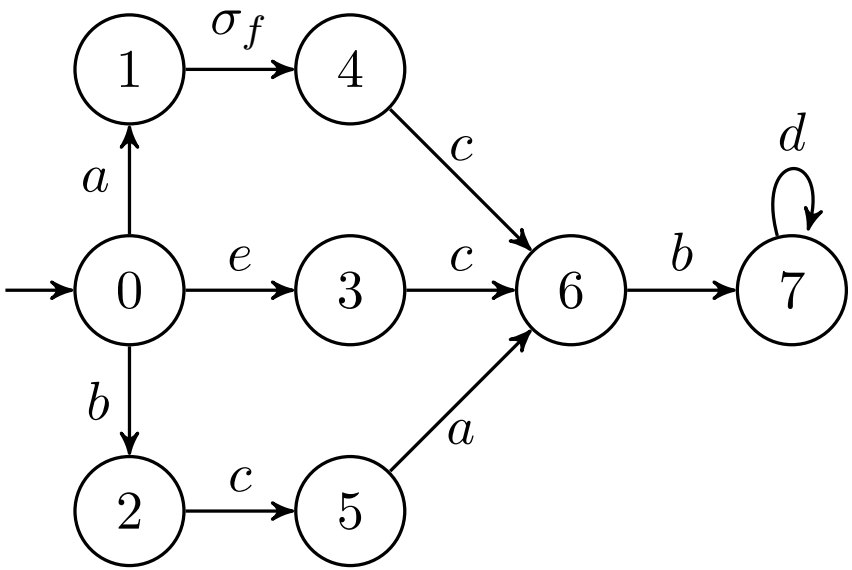
$$P_{\Sigma,\Sigma_o}(s_Y) = P_{\Sigma,\Sigma_o}(s_N) = cd^n$$

- $L(G)$ **not diagnosable**

Language diagnosability

$$G = (X, \Sigma, \delta, x_0)$$

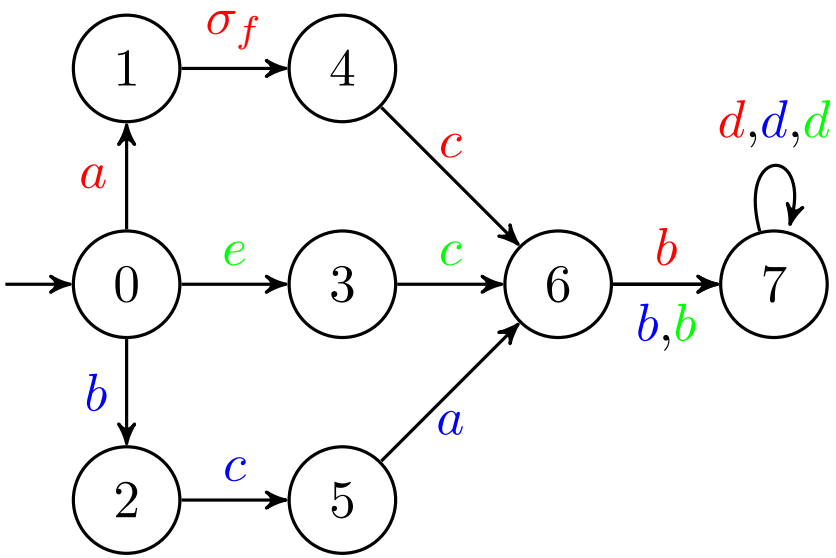
- $X = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{a, b, c, d, \sigma_f\}$
- $x_0 = 0$
- $\Sigma_o = \{b, d, e\}$; $\Sigma_{uo} = \{a, c, \sigma_f\}$
- $\Sigma_f = \{\sigma_f\}$



Language diagnosability

$$G = (X, \Sigma, \delta, x_0)$$

- $X = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{a, b, c, d, \sigma_f\}$
- $x_0 = 0$

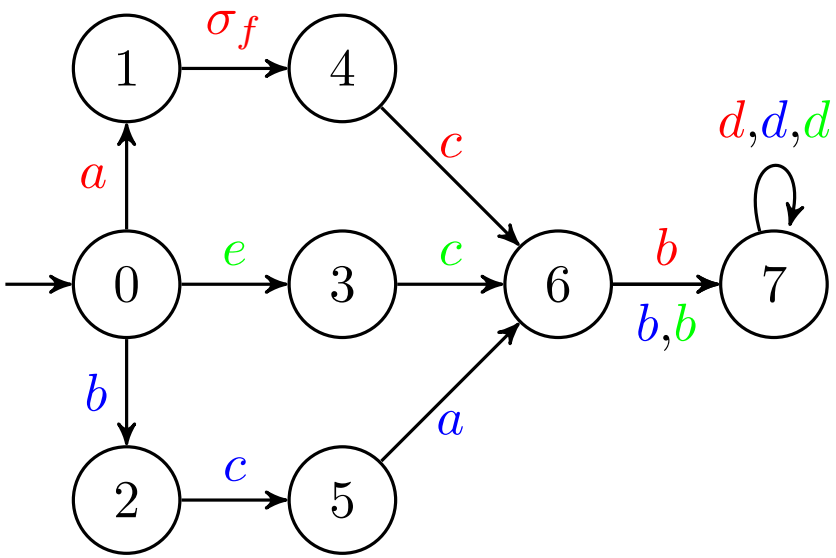


- $\Sigma_o = \{b, d, e\}$; $\Sigma_{uo} = \{a, c, \sigma_f\}$
- $\Sigma_f = \{\sigma_f\}$
- $s_Y = a\sigma_f cbd^n \Rightarrow P_{\Sigma,\Sigma_o}(s_Y) = bd^n$
- $s'_N = bcabd^n \Rightarrow P_{\Sigma,\Sigma_o}(s'_N) = bbd^n$
- $s''_N = ecbd^n \Rightarrow P_{\Sigma,\Sigma_o}(s''_N) = ebd^n$

Language diagnosability

$$G = (X, \Sigma, \delta, x_0)$$

- $X = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- $\Sigma = \{a, b, c, d, \sigma_f\}$
- $x_0 = 0$



- $\Sigma_o = \{b, d, e\}$; $\Sigma_{uo} = \{a, c, \sigma_f\}$
- $\Sigma_f = \{\sigma_f\}$
- $s_Y = a\sigma_f cbd^n \Rightarrow P_{\Sigma,\Sigma_o}(s_Y) = bd^n$
- $s'_N = bcabd^n \Rightarrow P_{\Sigma,\Sigma_o}(s'_N) = bbd^n$
- $s''_N = ecbd^n \Rightarrow P_{\Sigma,\Sigma_o}(s''_N) = ebd^n$
- **$L(G)$ diagnosable**

Idea Behind Language Diagnosability

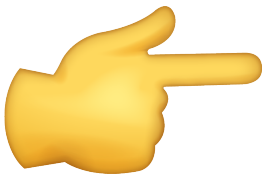
- Fault sequence: $s \in L$ is a faulty sequence if $\sigma_f \in s$.
- Normal sequence: if $\sigma_f \notin s$, then s is a normal sequence.

Idea Behind Language Diagnosability

- Fault sequence: $s \in L$ is a faulty sequence if $\sigma_f \in s$.
- Normal sequence: if $\sigma_f \notin s$, then s is a normal sequence.
- Ambiguous sequence: a fault sequence $s_Y \in L$ is an ambiguous sequence with respect to projection P_{Σ,Σ_o} and σ_f if there exists a normal sequence $s_N \in L$ such that $P_{\Sigma,\Sigma_o}(s_Y) = P_{\Sigma,\Sigma_o}(s_N)$.

Idea Behind Language Diagnosability

- Fault sequence: $s \in L$ is a faulty sequence if $\sigma_f \in s$.
- Normal sequence: if $\sigma_f \notin s$, then s is a normal sequence.
- Ambiguous sequence: a fault sequence $s_Y \in L$ is an ambiguous sequence with respect to projection P_{Σ,Σ_o} and σ_f if there exists a normal sequence $s_N \in L$ such that $P_{\Sigma,\Sigma_o}(s_Y) = P_{\Sigma,\Sigma_o}(s_N)$.



Diagnosability requires that there do NOT exist AMBIGUOUS SEQUENCES

Formal Definition of Language Diagnosability

- Post-language of L after s: $L/s = \{t \in \Sigma^* : st \in L\}$

Formal Definition of Language Diagnosability

- Post-language of L after s: $L/s = \{t \in \Sigma^* : st \in L\}$
- $\Psi(\Sigma_f) = \{s\sigma_f \in L : (s \in \Sigma^*) \wedge (\sigma_f \in \Sigma_f)\}$

Formal Definition of Language Diagnosability

- Post-language of L after s: $L/s = \{t \in \Sigma^* : st \in L\}$
- $\Psi(\Sigma_f) = \{s\sigma_f \in L : (s \in \Sigma^*) \wedge (\sigma_f \in \Sigma_f)\}$
- Language $L(G)$, is diagnosable with respect to projection P_{Σ,Σ_o} and $\Sigma_f = \{\sigma_f\}$ if the following holds true:

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall t \in L/s)(\|t\| \geq n \Rightarrow D),$$

where the diagnosability condition D is

$$(\forall \omega \in P^{-1}_{\Sigma,\Sigma_o}(P_{\Sigma,\Sigma_o}(st)) \cap L)(\Sigma_f \in \omega),$$

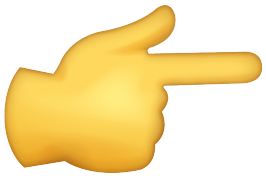
Formal Definition of Language Diagnosability

- Post-language of L after s: $L/s = \{t \in \Sigma^* : st \in L\}$
- $\Psi(\Sigma_f) = \{s\sigma_f \in L : (s \in \Sigma^*) \wedge (\sigma_f \in \Sigma_f)\}$
- Language $L(G)$, is diagnosable with respect to projection P_{Σ,Σ_o} and $\Sigma_f = \{\sigma_f\}$ if the following holds true:

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall t \in L/s)(\|t\| \geq n \Rightarrow D),$$

where the diagnosability condition D is

$$(\forall \omega \in P^{-1}_{\Sigma,\Sigma_o}(P_{\Sigma,\Sigma_o}(st)) \cap L)(\Sigma_f \in \omega),$$



Diagnosability requires that there do NOT exist AMBIGUOUS SEQUENCES

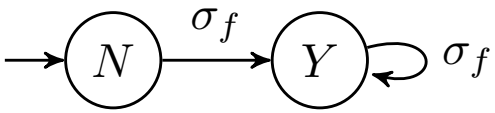
III. Online Diagnosis

Diagnoser Automaton

- $G = (X, \Sigma, \delta, x_0)$: system automaton whose language is diagnosable with respect to P_{Σ,Σ_o} and σ_f

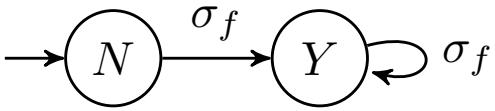
Diagnoser Automaton

- $G = (X, \Sigma, \delta, x_0)$: system automaton whose language is diagnosable with respect to P_{Σ,Σ_o} and σ_f
- Label automaton: $A_\ell = (X_\ell, \Sigma_f, \delta_\ell, x_{\ell_0})$



Diagnoser Automaton

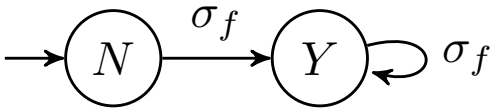
- $G = (X, \Sigma, \delta, x_0)$: system automaton whose language is diagnosable with respect to P_{Σ,Σ_o} and σ_f
- Label automaton: $A_\ell = (X_\ell, \Sigma_f, \delta_\ell, x_{\ell_0})$



- Labeled automaton: $G_\ell = G||A_\ell$

Diagnoser Automaton

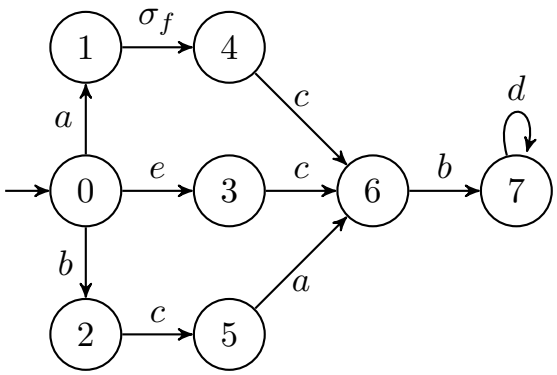
- $G = (X, \Sigma, \delta, x_0)$: system automaton whose language is diagnosable with respect to P_{Σ,Σ_o} and σ_f
- Label automaton: $A_\ell = (X_\ell, \Sigma_f, \delta_\ell, x_{\ell_0})$



- Labeled automaton: $G_\ell = G \| A_\ell$
- Diagnoser: $G_d(\Sigma_o) = \text{Obs}(G_\ell, \Sigma_o) = (X_d, \Sigma_o, \delta_d, x_{0,d})$

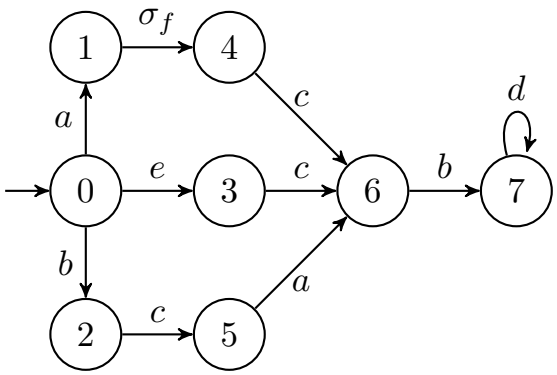
Diagnoser Automaton - Example

- System Automaton G

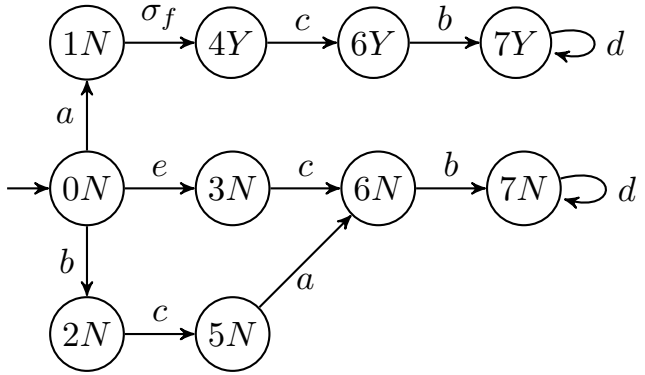


Diagnoser Automaton - Example

- System Automaton G

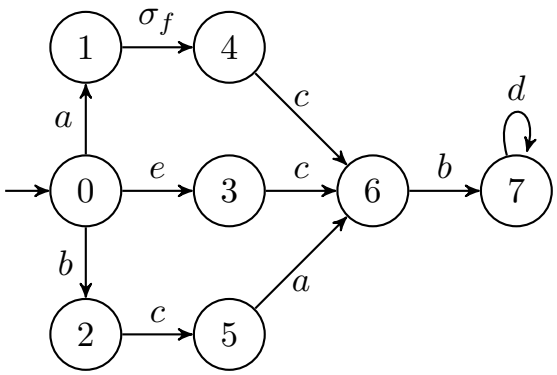


- Labeled Automaton G_ℓ

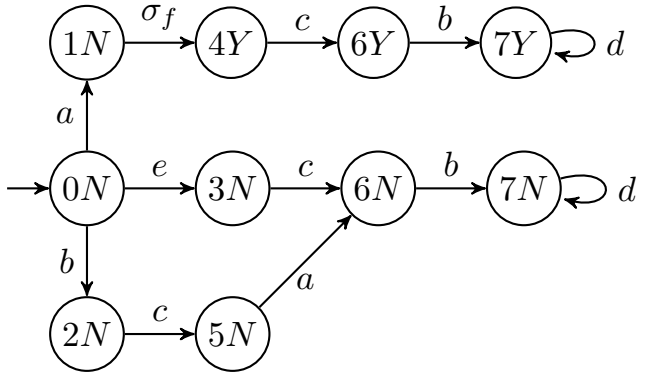


Diagnoser Automaton - Example

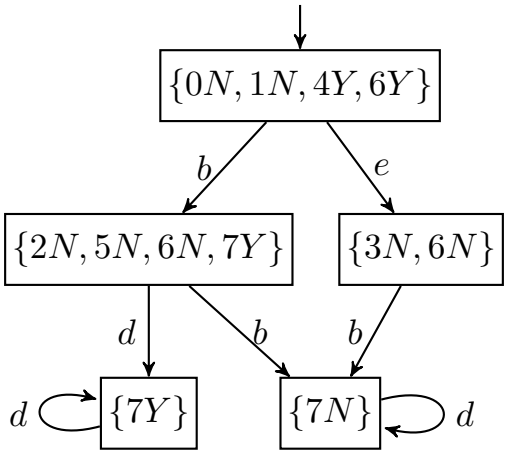
- System Automaton G



- Labeled Automaton G_ℓ



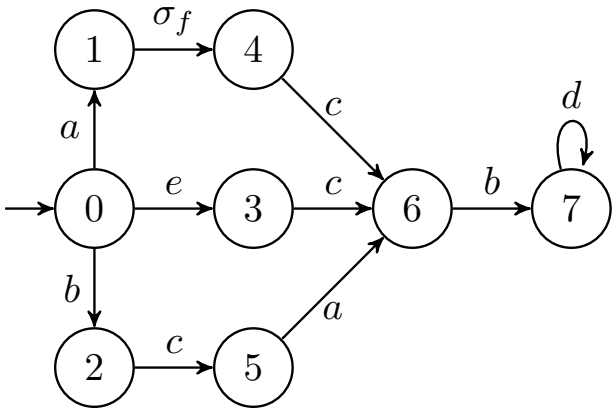
- Diagnoser Automaton G_d ($\Sigma_o = \{b, d, e\}$)



IV. Robust Fault Diagnosis of Discrete-Event Systems. Why?

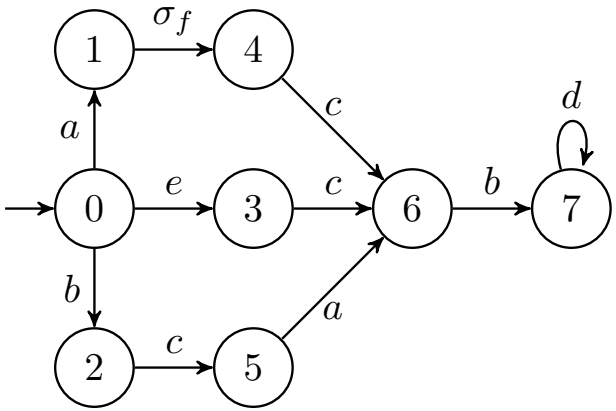
Motivation Example: loss of event observation

- System Automaton G

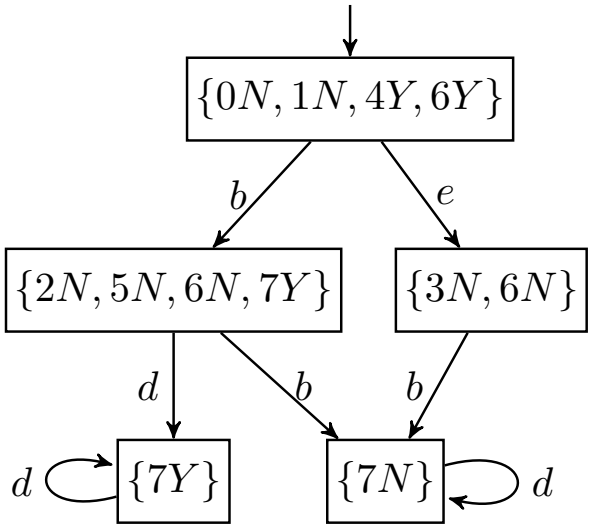


Motivation Example: loss of event observation

- System Automaton G

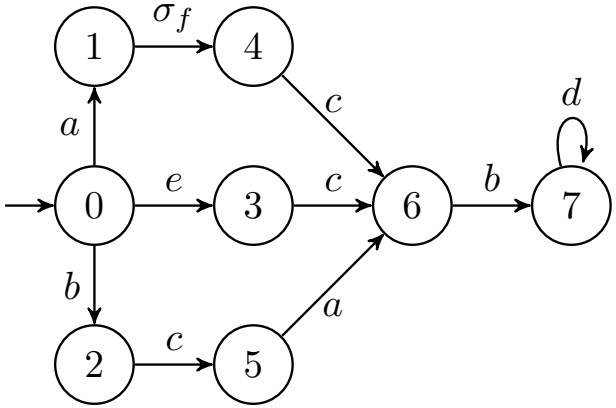


- Diagnoser G_d ($\Sigma_o = \{b, d, e\}$)

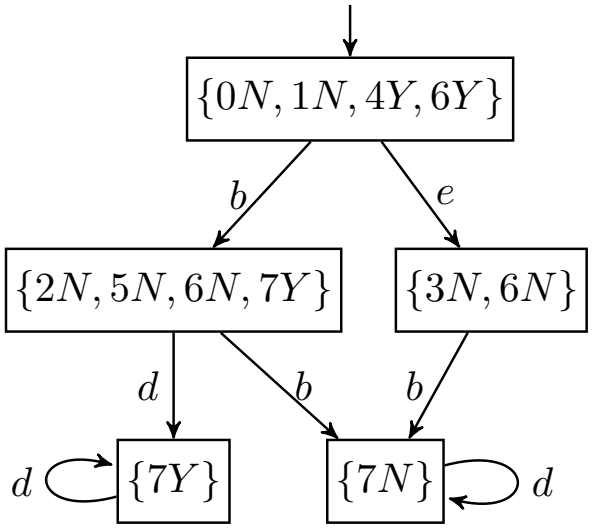


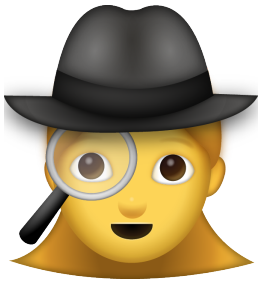
Motivation Example: loss of event observation

- System Automaton G



- Diagnoser G_d ($\Sigma_o = \{b, d, e\}$)

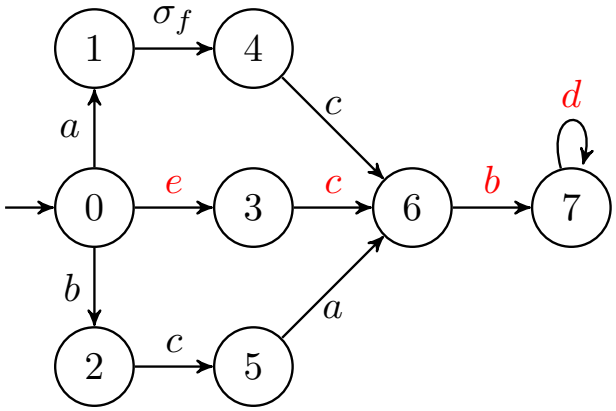




What would happen if, for some reason, the occurrence of event e did not reach G_d?

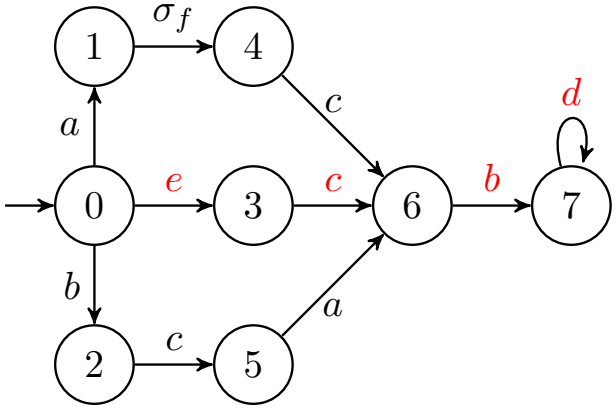
Motivation Example: loss of event observation

- System Automaton G

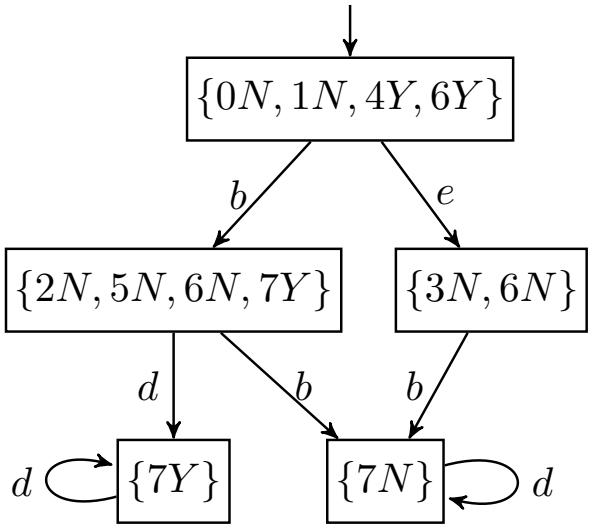


Motivation Example: loss of event observation

- System Automaton G

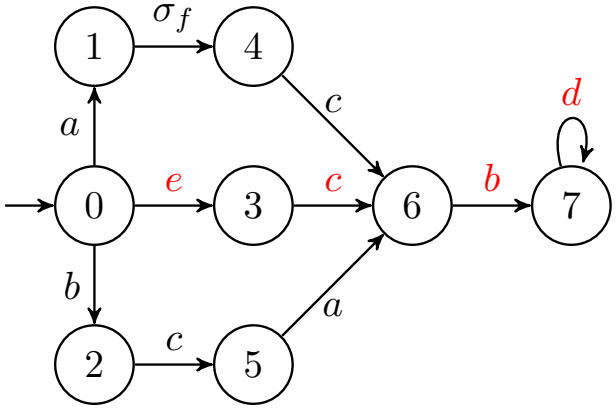


- Diagnoser G_d ($\Sigma_o = \{b, d, e\}$)

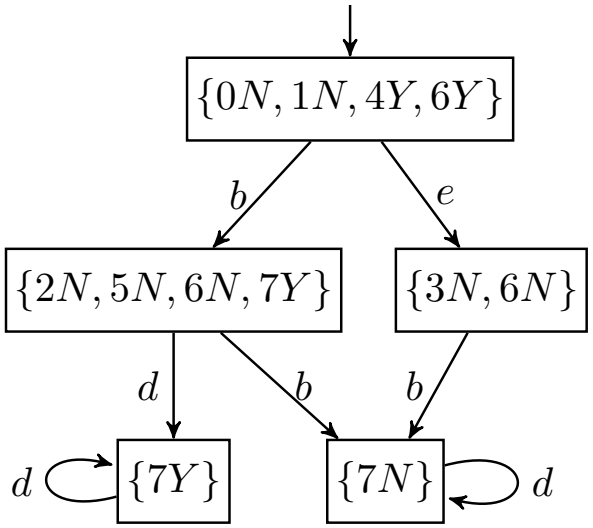


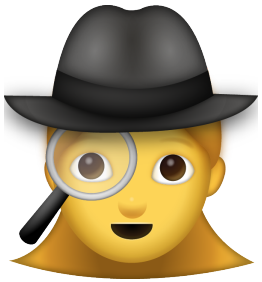
Motivation Example: loss of event observation

- System Automaton G



- Diagnoser G_d ($\Sigma_o = \{b, d, e\}$)

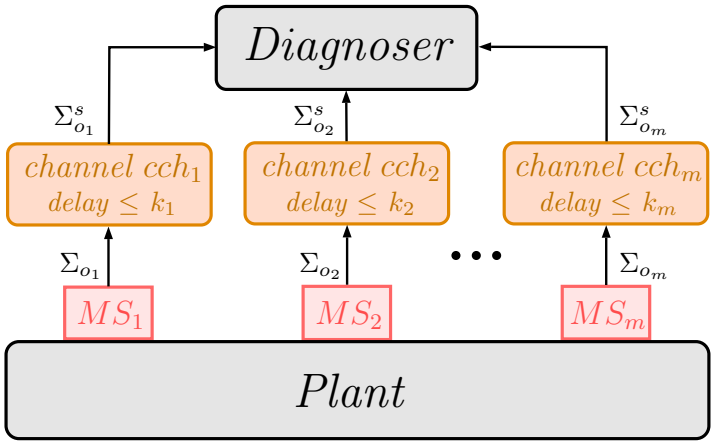




Diagnoser would end up in state $\{7Y\}$
FALSE POSITIVE

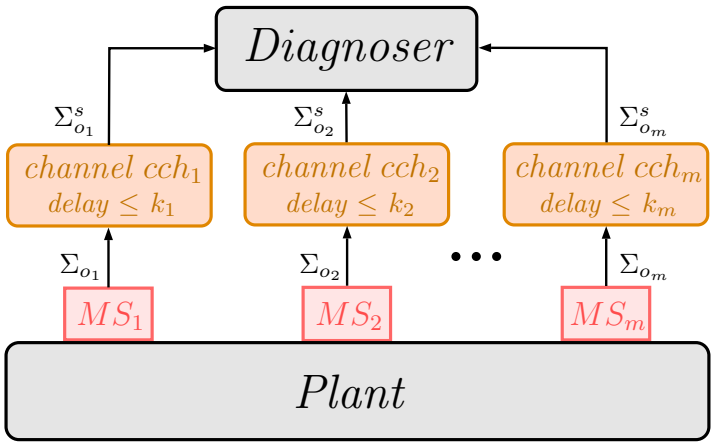
Motivation Example: Networked DES

- Networked



Motivation Example: Networked DES

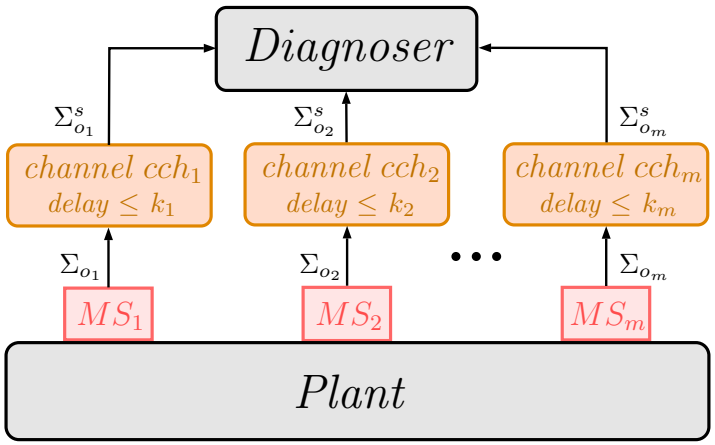
- Networked



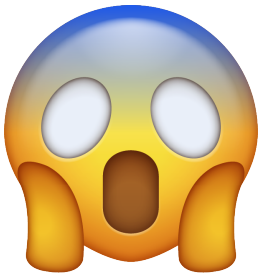
- Measurement sites: MS_i, $i = 1, 2, \ldots, m$
- Communication channels: cch_i, $i = 1, 2, \ldots, m$
- Communication channels delays: k_i, $i = 1, 2, \ldots, m$ Steps

Motivation Example: Networked DES

- Networked



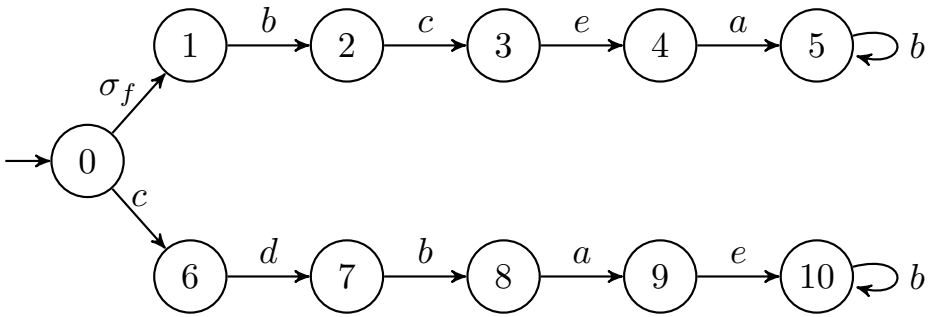
- Measurement sites: MS_i, $i = 1, 2, \ldots, m$
- Communication channels: cch_i, $i = 1, 2, \ldots, m$
- Communication channels delays: k_i, $i = 1, 2, \ldots, m$ Steps



Event observation can be performed in an order different from the actual event occurrences due to different delays of the various communication channels employed.

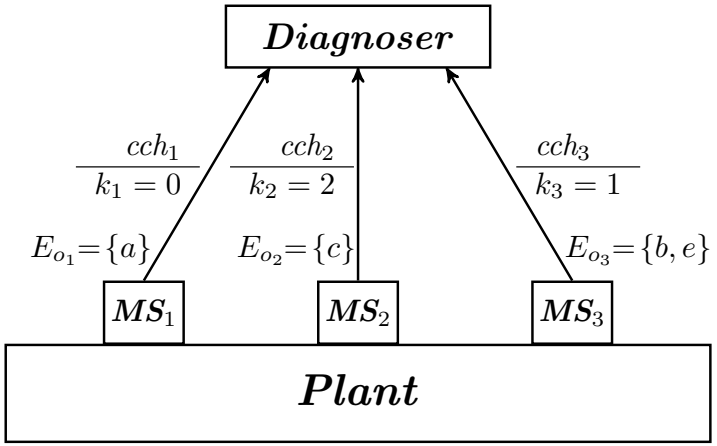
Motivation Example: Networked DES

- Automaton



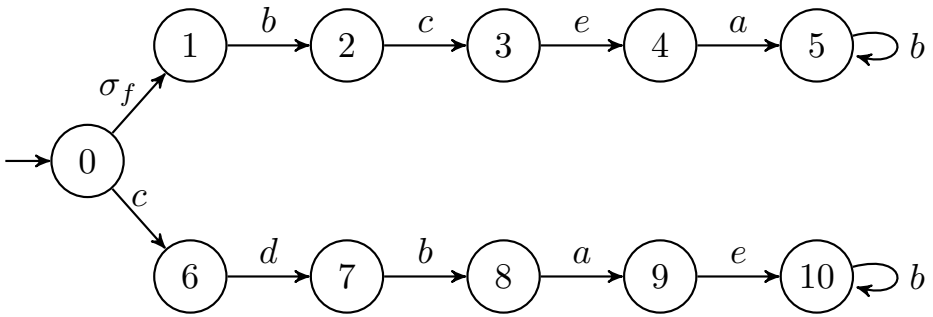
- $\Sigma = \{a, b, c, d, e, \sigma_f\}$, $\Sigma_f = \{\sigma_f\}$

- Networked DES



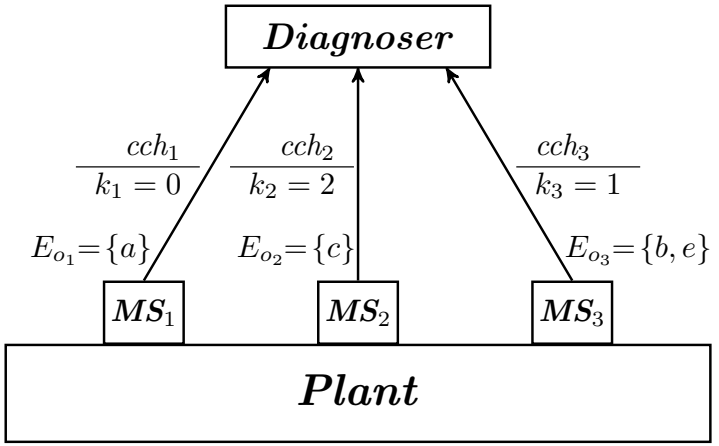
Motivation Example: Networked DES

- Automaton



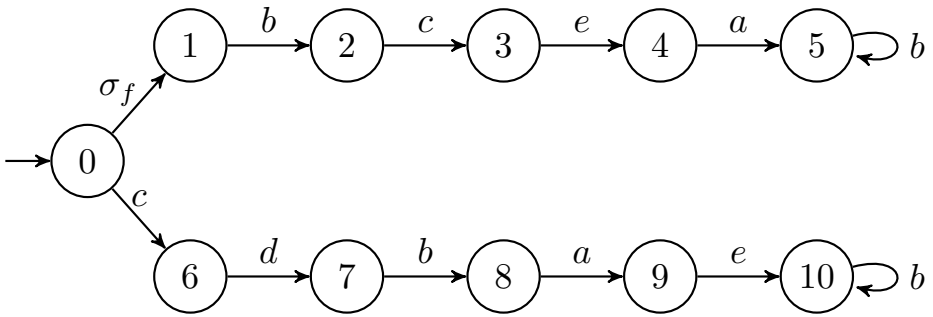
- $\Sigma = \{a, b, c, d, e, \sigma_f\}$, $\Sigma_f = \{\sigma_f\}$
  - Diagnosable if no delay exists

- Networked DES

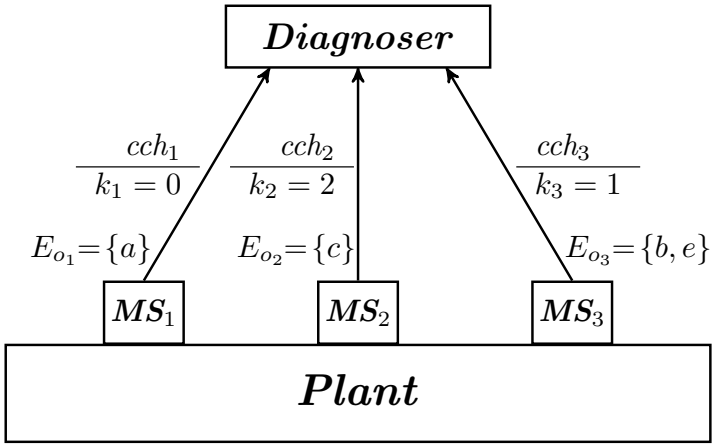


Motivation Example: Networked DES

- Automaton



- Networked DES



- $\Sigma = \{a, b, c, d, e, \sigma_f\}$, $\Sigma_f = \{\sigma_f\}$
  - Diagnosable if no delay exists
  - Assuming there are communication channel delays:
    - $s_Y = \sigma_f bceab^n$ and $s_N = cdbaeb^n$, $n \in \mathbb{Z}_+$
    - $s_{Y_a} = \sigma_f bcc_s b_s eaa_s e_s (bb_s)^n$, $s_{N_a} = cc_s dbb_s aa_s ee_s (bb_s)^n$
    - $P_{\Sigma_a, \Sigma_o^s}(s_{Y_a}) = P_{\Sigma_a, \Sigma_o^s}(s_{N_a}) = c_s b_s a_s e_s (b_s)^n$
    - Non diagnosable

V. Robust Diagnosis against intermittent loss of observation

Key to the Solution: Language augmentation

- Dilation (Carvalho et al., 2014)
  - $\Sigma_o = \Sigma_{ilo} \dot{\cup} \Sigma_{nilo}$

    Σ_{ilo}: subset of Σ_o whose events are subject to intermittent loss of observations

    Σ_{nilo}: set of events whose observations are never lost.
  - $\Sigma'_{ilo} = \{\sigma' : \sigma \in \Sigma_{ilo}\}$
  - $\Sigma_{dil} = \Sigma \cup \Sigma'_{ilo}$

Key to the Solution: Language augmentation

- Dilation (Carvalho et al., 2014)
  - $\Sigma_o = \Sigma_{ilo} \dot{\cup} \Sigma_{nilo}$

    Σ_{ilo}: subset of Σ_o whose events are subject to intermittent loss of observations

    Σ_{nilo}: set of events whose observations are never lost.
  - $\Sigma'_{ilo} = \{\sigma' : \sigma \in \Sigma_{ilo}\}$
  - $\Sigma_{dil} = \Sigma \cup \Sigma'_{ilo}$
  - The dilation D is the mapping

$$D : \Sigma \to 2^{\Sigma_{dil}}$$
$$\sigma \mapsto D(\sigma) = \begin{cases} \{\sigma\}, & \sigma \in \Sigma \setminus \Sigma_{ilo}, \\ \{\sigma, \sigma'\}, & \sigma \in \Sigma_{ilo}, \end{cases}$$

Key to the Solution: Language augmentation

- Dilation (Carvalho et al., 2014)
  - $\Sigma_o = \Sigma_{ilo} \dot{\cup} \Sigma_{nilo}$

    Σ_{ilo}: subset of Σ_o whose events are subject to intermittent loss of observations

    Σ_{nilo}: set of events whose observations are never lost.
  - $\Sigma'_{ilo} = \{\sigma' : \sigma \in \Sigma_{ilo}\}$
  - $\Sigma_{dil} = \Sigma \cup \Sigma'_{ilo}$
  - The dilation D is the mapping

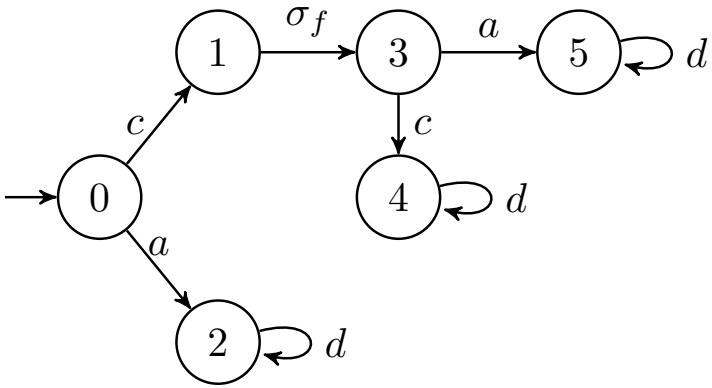
$$\begin{array}{rcccl} D & : & \Sigma & \to & 2^{\Sigma_{dil}} \\ & & \sigma & \mapsto & D(\sigma) = \left\{ \begin{array}{ll} \{\sigma\}, & \sigma \in \Sigma \setminus \Sigma_{ilo}, \\ \{\sigma, \sigma'\}, & \sigma \in \Sigma_{ilo}, \end{array} \right. \end{array}$$

  - Extension to languages:

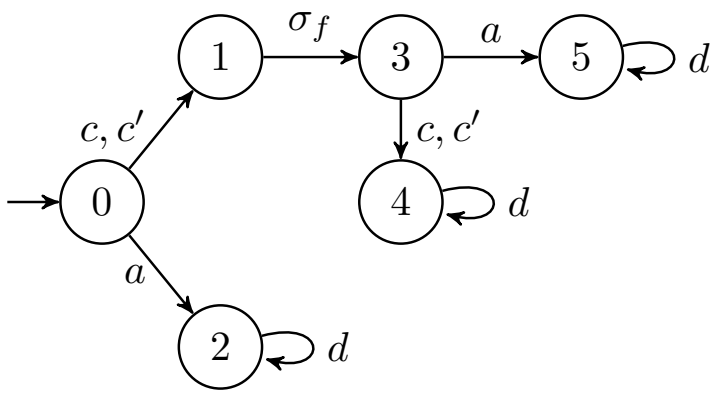
    $D(\varepsilon) = \varepsilon$, and $D(s\sigma) = D(s)D(\sigma)$, for $s \in \Sigma^*$ and $\sigma \in \Sigma$.

Dilation - Example

- Automaton G



- Dilated automaton G_{dil}



- $\Sigma = \{a, b, c, d, \sigma_f\}$, $\Sigma_o = \{a, b, c, d\}$, $\Sigma_f = \{\sigma_f\}$
- $L(G) = pre(c\sigma_f(a + c)d^* + ad^*)$
- $\Sigma_{ilo} = \{c\}$
- $L(G) = pre\big((c + c')\sigma_f(a + c + c')d^* + ad^*\big)$

Robust diagnosability

Diagnosability requires that there do NOT exist AMBIGUOUS SEQUENCES

Robust diagnosability

Diagnosability requires that there do NOT exist AMBIGUOUS SEQUENCES

- Definition: Language $L(G)$ is robustly diagnosable with respect to dilation D, projection $P_{\Sigma_{dil},\Sigma_o} : \Sigma^*_{dil} \to \Sigma^*_o$ and $\Sigma_f = \{\sigma_f\}$ if the following holds true:

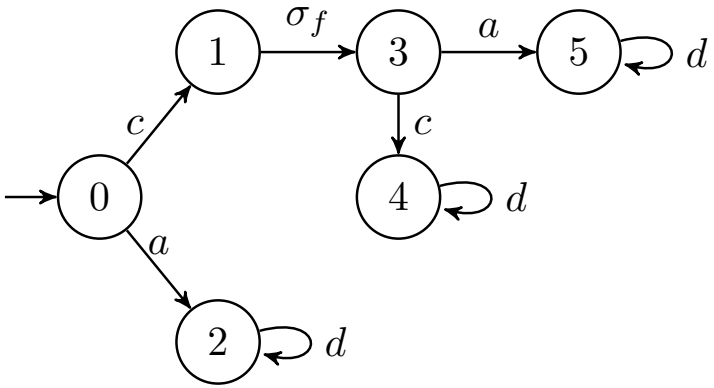
$$(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall t \in L/s)(\|t\| \geq n \Rightarrow R_D),$$

where the robust diagnosability condition R_D is

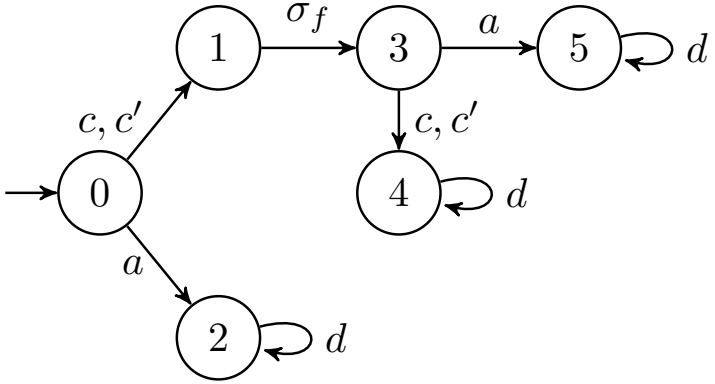
$$\left(\nexists \omega \in L \right) \left[P_{\Sigma_{dil},\Sigma_o}(D(st)) \cap P_{\Sigma_{dil},\Sigma_o}(D(\omega)) \neq \emptyset) \wedge (\Sigma_f \notin \omega) \right].$$

Robust Diagnosability - Example

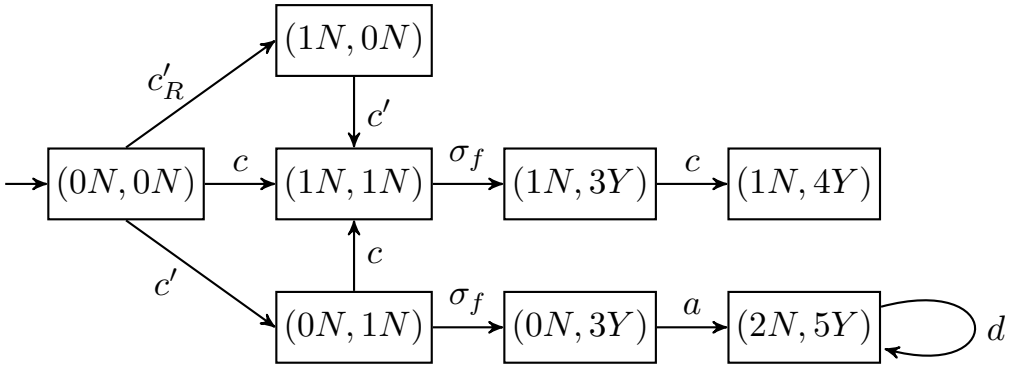
- Automaton G



- Dilated automaton G_{dil}

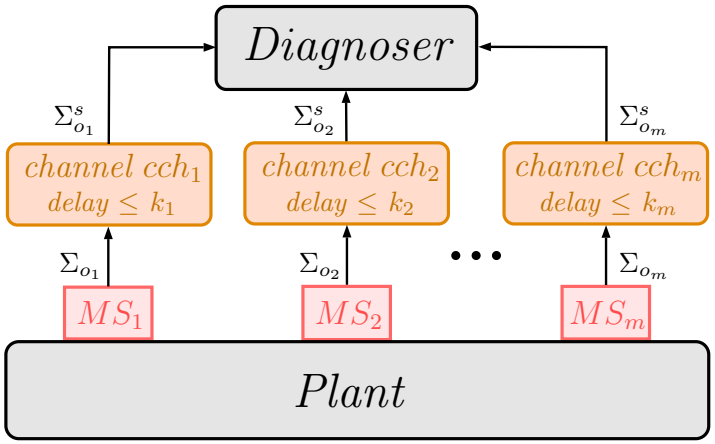


- Robust Diagnosability verification



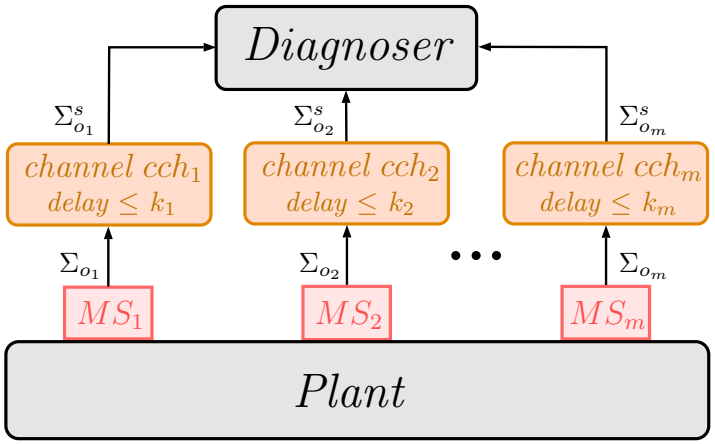
VI. Diagnosability of Networked DES

Networked DES



- Automaton:
  $G = (X, \Sigma, \delta, x_0)$
- Measurement sites:
  MS_i, $i = 1, \ldots, m$
- Channel delay structure:
  $\vec{k} = [\ k_1 \quad k_2 \quad \cdots \quad k_m\ ]$

Networked DES



- Automaton:
  $G = (X, \Sigma, \delta, x_0)$
- Measurement sites:
  MS_i, $i = 1, \ldots, m$
- Channel delay structure:
  $\vec{k} = [\ k_1 \quad k_2 \quad \cdots \quad k_m\ ]$

- Event occurrences and observations must be distinguished
  $\Sigma^s_{o_i} = \{\sigma_s : \sigma \in \Sigma_{o_i}\} \rightarrow \Sigma^s_o = \cup_{i=1}^m \Sigma^s_{o_i}$
- Augmented event set
  $\Sigma_a = \Sigma \cup \Sigma^s_o$
- Observation of system evolution
  $P_{\Sigma_a, \Sigma^s_o} : \Sigma_a^* \rightarrow \Sigma_o^{s^*}$.

Language augmentation (Nunes et al, 2018)

$$\chi \quad : \quad \Sigma^* \to 2^{\Sigma_a^*}$$

$$s \mapsto \chi(s) = \{s_a \in \Sigma_a^* : (s_a \models \textbf{C1}) \wedge (s_a \models \textbf{C2}) \wedge (s_a \models \textbf{C3})\}$$

- C1. $P_{\Sigma_a,\Sigma}(s_a) = s$;
- C2. For all $\sigma \in \Sigma_{o,i}$, if $\sigma_s^{(p)} \in s_a$, then:

$$\|P_{\Sigma_a,\Sigma_{o_i}^s}(pre(s_a, \sigma_s^{(p)}))\| - \|P_{\Sigma,\Sigma_{o_i}}(pre(s, \sigma^{(p)}))\| \leq k_i,$$

- C3. For all $\sigma_s \in \Sigma_{o,i}^s$, if $\sigma_s^{(p)} \in s_a$ then

$$\left(\sigma^{(p)} \in pre(s_a, \sigma_s^{(p)})\right) \wedge$$

$$\left(\|P_{\Sigma_a,\Sigma_{o_i}^s}(pre(s_a, \sigma_s^{(p)}))\| = \|P_{\Sigma,\Sigma_{o_i}}(pre(s, \sigma^{(p)}))\|\right)$$

The extension of χ to the domain 2^{Σ^*} is $\chi(L) := \bigcup_{t \in L} \chi_i(t)$.

Network diagnosability

Diagnosability requires that there do NOT exist AMBIGUOUS SEQUENCES

Network diagnosability

Diagnosability requires that there do NOT exist AMBIGUOUS SEQUENCES

Definition: Language $L(G)$ is network diagnosable with respect to augmentation $\chi : 2^{\Sigma^*} \to 2^{\Sigma_a^*}$, projection $P_{\Sigma_a,\Sigma_o^s} : \Sigma_a^* \to \Sigma_o^{s^*}$ and $\Sigma_f = \{\sigma_f\}$ if the following holds true:

$$(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall t \in L/s)(||t|| \geq n \Rightarrow N_D)$$

where the network diagnosability condition N_D is

$$(\nexists w \in L)\left[P_{\Sigma_a,\Sigma_o^s}(\chi(st)) \cap P_{\Sigma_a,\Sigma_o^s}(\chi(w)) \neq \emptyset) \wedge (\Sigma_f \notin w)\right].$$

Verification of Network diagnosability

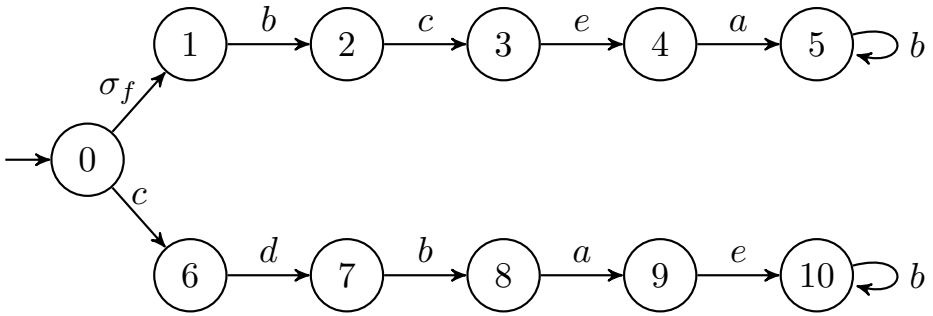
- Basic idea: construct an augmented automaton G_a such that $L(G_a) = \chi(L(G))$

Verification of Network diagnosability

- Basic idea: construct an augmented automaton G_a such that $L(G_a) = \chi(L(G))$
- Alves et al., (2021) proposes a construction for $G_a = (X_a, \Sigma_a, \delta_a, x_{a_0})$, whose general idea is as follows:
  - $x_a = (x, q)$
    ✓ x is the current state of G
    ✓ $q = \sigma_1 n_1 \sigma_1 n_2 \ldots \sigma_p n_p$
    where
    ✓ $\sigma_i \in \Sigma_o$: event occurrence
    ✓ n_i : counts the number of event occurrences (observable or unobservable) after the occurrence of σ_i
  - $x_a = (5, c2a0)$
    ✓ G is currently at 5 and ✓ either one unobservable event has occurred between the occurrences of c and a, ✓ or one observable event has occurred and its transmission has been successfully received at the diagnoser.

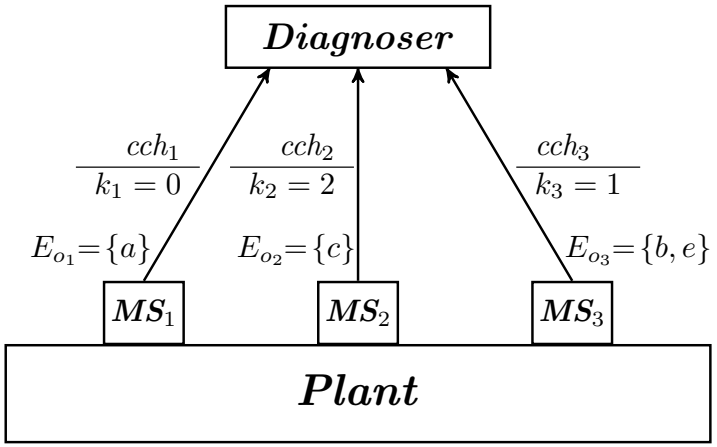
Back to the Motivation Example of Networked DES

- Automaton

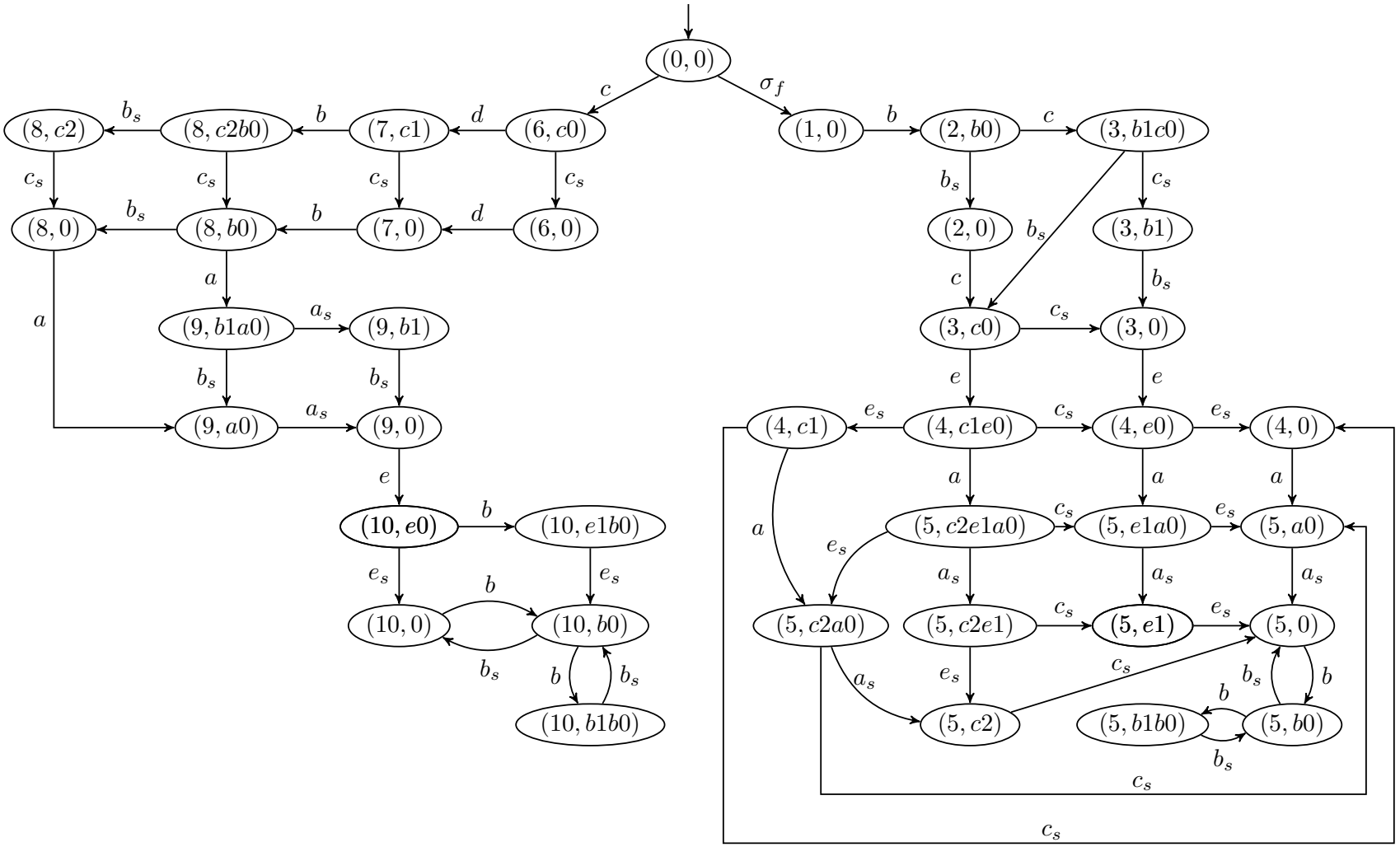


- $\Sigma = \{a, b, c, d, e, \sigma_f\}$, $\Sigma_f = \{\sigma_f\}$
- $\Sigma_o^s = \{a_s, b_s, c_s, e_s\}$
- $\vec{k} = [\ 0 \quad 2 \quad 1\ ]$

- Networked DES



Augmented automaton G_a



Verification of network diagnosability

- It can be performed with any verification algorithm applied on G_a assuming Σ_o^s as the set of observable events (Moreira et al., 2012, Viana & Basilio, 2019)

Verification of network diagnosability

- It can be performed with any verification algorithm applied on G_a assuming Σ_o^s as the set of observable events (Moreira et al., 2012, Viana & Basilio, 2019)
- Loss of observation can also be taken into account by applying dilation to events $\sigma_s \in \Sigma_o^s$ that are subject to intermittent loss of observation

Verification of network diagnosability

- It can be performed with any verification algorithm applied on G_a assuming Σ_o^s as the set of observable events (Moreira et al., 2012, Viana & Basilio, 2019)
- Loss of observation can also be taken into account by applying dilation to events $\sigma_s \in \Sigma_o^s$ that are subject to intermittent loss of observation
- $L(G)$ is not network diagnosable with respect to χ, P_{Σ_a,Σ_o^s} and σ_f.

VII. Conclusion

Concluding remarks

- We have restricted ourselves to the problem of diagnosability and diagnosis of monolithic DES

Concluding remarks

- We have restricted ourselves to the problem of diagnosability and diagnosis of monolithic DES
- When the physical system has a distributed structure, it is more appropriate to consider the decentralized diagnosability notions proposed in Debouk at al. (2000) and Contant et al. (2002)

Concluding remarks

- We have restricted ourselves to the problem of diagnosability and diagnosis of monolithic DES
- When the physical system has a distributed structure, it is more appropriate to consider the decentralized diagnosability notions proposed in Debouk at al. (2000) and Contant et al. (2002)
- Indeed, robust diagnosability was introduced in the DES community for decentralized DES (Basilio and Lafortune, 2009)

Concluding remarks

- We have restricted ourselves to the problem of diagnosability and diagnosis of monolithic DES
- When the physical system has a distributed structure, it is more appropriate to consider the decentralized diagnosability notions proposed in Debouk at al. (2000) and Contant et al. (2002)
- Indeed, robust diagnosability was introduced in the DES community for decentralized DES (Basilio and Lafortune, 2009)
- The augmentation approach adopted in the diagnosability of networked DES can be leveraged so as to allow other problems of networked DES, such as networked supervisory control or opacity enforcement, to be converted in standard problems, which can be solved with existing tools.

Concluding remarks

- We have restricted ourselves to the problem of diagnosability and diagnosis of monolithic DES
- When the physical system has a distributed structure, it is more appropriate to consider the decentralized diagnosability notions proposed in Debouk at al. (2000) and Contant et al. (2002)
- Indeed, robust diagnosability was introduced in the DES community for decentralized DES (Basilio and Lafortune, 2009)
- The augmentation approach adopted in the diagnosability of networked DES can be leveraged so as to allow other problems of networked DES, such as networked supervisory control or opacity enforcement, to be converted in standard problems, which can be solved with existing tools.
- Robust diagnosis of DES is a lively research topic. There is still much to be done!

Acknowledgments

- Brazilian Ministry of Education, CAPES, Finance code 001

Acknowledgments

- Brazilian Ministry of Education, CAPES, Finance code 001
- Brazilian Research Council (CNPq), grants 309.652/2017-0 and 436.672/2018-9
- Rong Su, for inviting me to give this talk

Acknowledgments

- Brazilian Ministry of Education, CAPES, Finance code 001
- Brazilian Research Council (CNPq), grants 309.652/2017-0 and 436.672/2018-9
- Rong Su, for inviting me to give this talk
- Stéphane Lafortune, Marcos Moreira, Lilian Carvalho, Gustavo Viana, Marcos Vinícius Alves, Carlos Eduardo Nunes

Acknowledgments

- Brazilian Ministry of Education, CAPES, Finance code 001
- Brazilian Research Council (CNPq), grants 309.652/2017-0 and 436.672/2018-9
- Rong Su, for inviting me to give this talk
- Stéphane Lafortune, Marcos Moreira, Lilian Carvalho, Gustavo Viana, Marcos Vinícius Alves, Carlos Eduardo Nunes
- My son Renan Basilio, for helping me in the production of this presentation

Bibliography

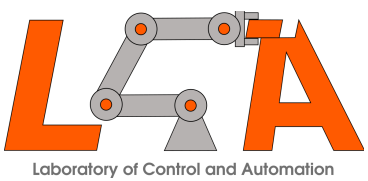
- Alves, M. V. S, Carvalho, L. K. & Basilio, J. C. (2021). Supervisory control of networked discrete event systems with timing structure. *IEEE Transactions on Automatic Control* (to appear).
- Basilio, J. C., & Lafortune, S. (2009). Robust codiagnosability of discrete event systems. in *Proceedings of the 2009 American Control Conference* (pp. 2202-2209).
- Carvalho, L. K., Basilio, J. C., & Moreira, M. V. (2012). Robust diagnosis of discrete event systems against intermittent loss of observations. *Automatica*, 48(9), 2068-2078.
- Contant, O., Lafortune, S., & Teneketzis, D. (2006). Diagnosability of discrete event systems with modular structure. *Discrete Event Dynamic Systems*, 16(1), 9-37.
- Debouk, R., Lafortune, S., & Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems*, 10(1-2), 33-86.
- Moreira, M. V., Jesus, T. C., & Basilio, J. C. (2011). Polynomial time verification of decentralized diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 56(7), 1679-1684.
- Nunes, C. E., Moreira, M. V., Alves, M. V., Carvalho, L. K., & Basilio, J. C. (2018). Codiagnosability of networked discrete event systems subject to communication delays and intermittent loss of observation. *Discrete Event Dynamic Systems*, 28(2), 215-246.
- Viana, G. S., & Basilio, J. C. (2019). Codiagnosability of discrete event systems revisited: A new necessary and sufficient condition and its applications. *Automatica*, 101, 354-364.

Robust Failure Diagnosis of Discrete Event Systems and Its Applications

João Carlos Basilio

basilio@dee.ufrj.br

Workshop on Analysis and Control for Resilience of Discrete Event Systems



Universidade Federal do Rio de Janeiro