Sensor Minimization Problems with Static or Dynamic Observers for Fault Diagnosis

Franck Cassez
CNRS/IRCCyN
Nantes, France

Stavros Tripakis
Cadence Research
Berkeley, USA
and
CNRS/VERIMAG
Grenoble, France

Karine Altisen
INPG/VERIMAG
Grenoble, France

7th Conf. on Applications of Concurrency
to System Design (ACSD'07)
Bratislava

July 2007

Outline of the talk

- Fault Diagnosis for Finite State Systems
  - Fault Diagnosis Problem
  - Minimization of a Static Observer
  - Minimization of Masks
- Fault Diagnosis with Dynamic Observers
  - Dynamic Observers
  - Diagnosability with Dynamic Observers
  - Synthesis of Dynamic Observers

Outline of the talk

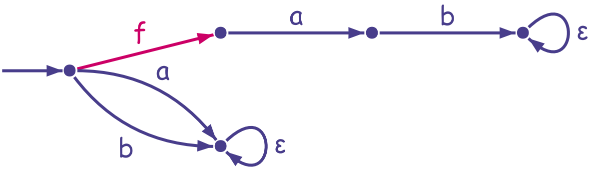
- Fault Diagnosis for Finite State Systems
  - Fault Diagnosis Problem
  - Minimization of a Static Observer
  - Minimization of Masks

- Fault Diagnosis with Dynamic Observers
  - Dynamic Observers
  - Diagnosability with Dynamic Observers
  - Synthesis of Dynamic Observers

Outline

- Fault Diagnosis for Finite State Systems
  - Fault Diagnosis Problem
  - Minimization of a Static Observer
  - Minimization of Masks
- Fault Diagnosis with Dynamic Observers

Fault Diagnosis

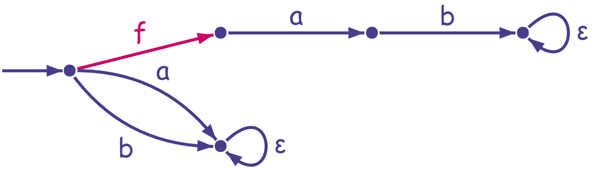


- A finite automaton A over $\Sigma^{\varepsilon,f} = \Sigma \cup \{\varepsilon, f\}$
- f is the fault action, Σ is the set of observable events
- k-faulty run contain f followed by more than k actions

  Faulty$_{\geq k}(A)$
- Non faulty run: contains no f

  NonFaulty(A)

Fault Diagnosis



- A finite automaton A over $\Sigma^{\varepsilon,f} = \Sigma \cup \{\varepsilon, f\}$
- f is the fault action, Σ is the set of observable events
- k-faulty run contain f followed by more than k actions

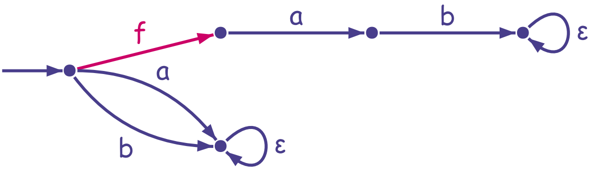
Faulty$_{\geq k}(A)$

- Non faulty run: contains no f

NonFaulty(A)

> Aim: observe Σ^* sequences and detect k-faulty runs

Fault Diagnosis



- A finite automaton A over $\Sigma^{\varepsilon,f} = \Sigma \cup \{\varepsilon, f\}$
- f is the fault action, Σ is the set of observable events
- k-faulty run contain f followed by more than k actions

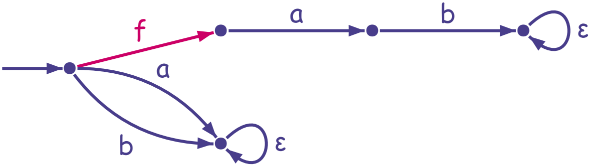
  Faulty$_{\geq k}(A)$
- Non faulty run: contains no f

  NonFaulty(A)

Role of an observer:

- never raise an alarm on non-faulty runs
- must raise an alarm on k-faulty runs

Diagnosis Problem



$tr(\rho)$ = trace of the run ρ (it is a word in $(\Sigma \cup \{f\})^*$)
$\pi_{/\Sigma}(tr(\rho))$ = projection of the trace of the run on observable events

Definition (k-diagnoser)

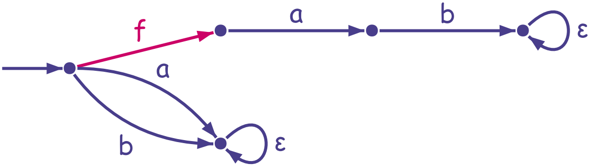
A mapping $D : \Sigma^* \longrightarrow \{0,1\}$ is a k-diagnoser for A if:

- for each run $\rho \in$ **NonFaulty**(A), $D(\pi_{/\Sigma}(tr(\rho))) = 0$;
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(\pi_{/\Sigma}(tr(\rho))) = 1$.

(Σ, k)-Diagnosability Problem

Given $A, \Sigma, k \in \mathbb{N}$, is there a k-diagnoser for A?

Diagnosis Problem



$tr(\rho)$ = trace of the run ρ (it is a word in $(\Sigma \cup \{f\})^*$)
$\pi_{/\Sigma}(tr(\rho))$ = projection of the trace of the run on observable events

Definition (k-diagnoser)

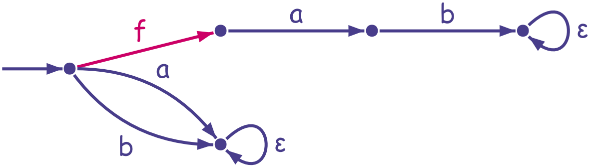
A mapping $D : \Sigma^* \longrightarrow \{0,1\}$ is a k-diagnoser for A if:

- for each run $\rho \in$ **NonFaulty**(A), $D(\pi_{/\Sigma}(tr(\rho))) = 0$;
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(\pi_{/\Sigma}(tr(\rho))) = 1$.

(Σ, k)-Diagnosability Problem

Given $A, \Sigma, k \in \mathbb{N}$, is there a k-diagnoser for A?

Diagnosis Problem



$tr(\rho)$ = trace of the run ρ (it is a word in $(\Sigma \cup \{f\})^*$)
$\pi_{/\Sigma}(tr(\rho))$ = projection of the trace of the run on observable events

Definition (k-diagnoser)

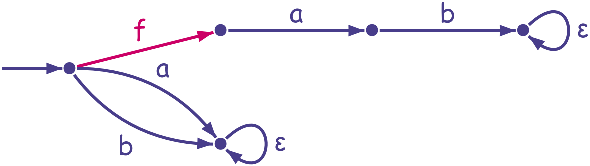
A mapping $D : \Sigma^* \longrightarrow \{0, 1\}$ is a k-diagnoser for A if:

- for each run $\rho \in$ **NonFaulty**(A), $D(\pi_{/\Sigma}(tr(\rho))) = 0$;
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(\pi_{/\Sigma}(tr(\rho))) = 1$.

(Σ, k)-Diagnosability Problem

Given A, Σ, $k \in \mathbb{N}$, is there a k-diagnoser for A?

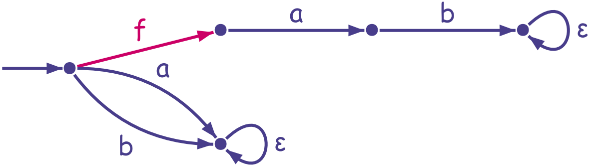
Example



- $\Sigma = \{a, b\}$. Is the plant 1, 2-diagnosable?
- $\Sigma = \{b\}$. Is the plant 1, 2, k-diagnosable?

k-Diagnosability $\iff$ $tr(\textbf{Faulty}_{\geq k}(A)) \cap tr(\textbf{NonFaulty}(A)) = \emptyset$

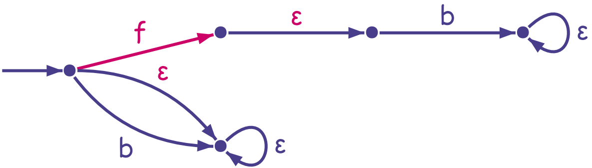
Example



- $\Sigma = \{a, b\}$. Is the plant 1, 2-diagnosable?
- $\Sigma = \{b\}$. Is the plant 1, 2, k-diagnosable?

k-Diagnosability $\iff$ $tr(\textbf{Faulty}_{\geq k}(A)) \cap tr(\textbf{NonFaulty}(A)) = \emptyset$

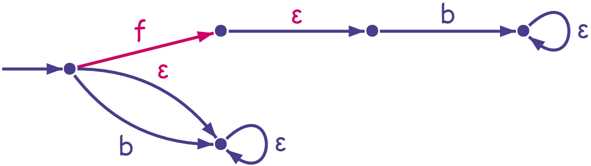
Example



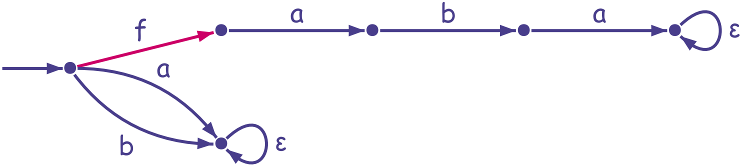
- $\Sigma = \{a, b\}$. Is the plant 1, 2-diagnosable? Yes
- $\Sigma = \{b\}$. Is the plant 1, 2, k-diagnosable?

k-Diagnosability $\iff$ $tr(\textbf{Faulty}_{\geq k}(A)) \cap tr(\textbf{NonFaulty}(A)) = \emptyset$

Example



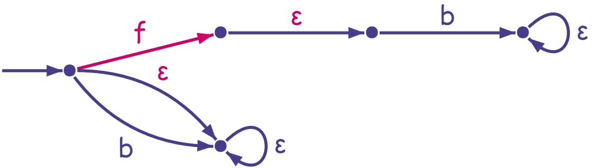
- $\Sigma = \{a, b\}$. Is the plant 1, 2-diagnosable? Yes
- $\Sigma = \{b\}$. Is the plant 1, 2, k-diagnosable? No



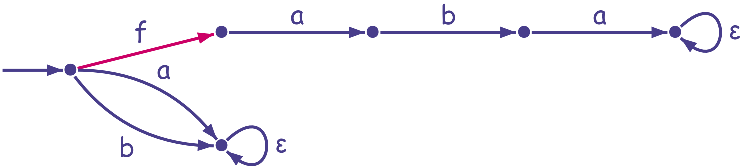
- $\Sigma = \{a\}$. Is the plant 1, 2-diagnosable?

k-Diagnosability $\iff$ $tr(\textbf{Faulty}_{\geq k}(A)) \cap tr(\textbf{NonFaulty}(A)) = \emptyset$

Example



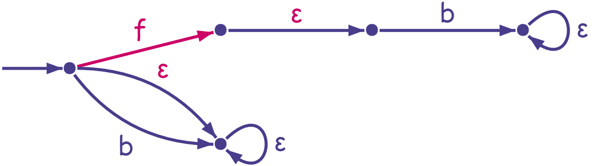
- $\Sigma = \{a, b\}$. Is the plant 1, 2-diagnosable? *Yes*
- $\Sigma = \{b\}$. Is the plant 1, 2, k-diagnosable? No



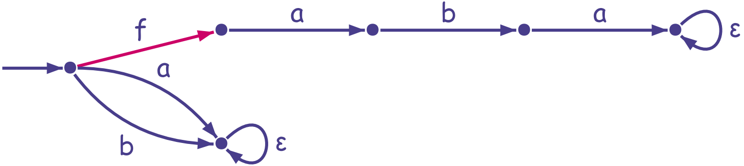
- $\Sigma = \{a\}$. Is the plant 1, 2-diagnosable?

k-Diagnosability $\iff$ $tr(\textbf{Faulty}_{\geq k}(A)) \cap tr(\textbf{NonFaulty}(A)) = \emptyset$

Example



- $\Sigma = \{a, b\}$. Is the plant 1, 2-diagnosable? Yes
- $\Sigma = \{b\}$. Is the plant 1, 2, k-diagnosable? No



- $\Sigma = \{a\}$. Is the plant 1, 2-diagnosable? …3-diagnosable

k-Diagnosability $\Longleftrightarrow$ $tr(\mathbf{Faulty}_{\geq k}(A)) \cap tr(\mathbf{NonFaulty}(A)) = \emptyset$

Results for the Diagnosis Problem

[Sampath et al., IEEE TAC 1995, Jiang et al., IEEE TAC 2001]

A is (Σ, k)-diagnosable if there is a k-diagnoser for A.

A is Σ-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is (Σ, k)-diagnosable.

Diagnosis Problems

- (A) Is A Σ-diagnosable ?
- (B) If "yes" to (A), compute the minimum k and
- (C) Compute a witness diagnoser.

Results for Diagnosis Problem

- (A) is in PTIME
- (B) is in PTIME
- (C) is in EXPTIME

A witness diagnoser is an automaton with at most $2^{O(A)}$ states

Results for the Diagnosis Problem

[Sampath et al., IEEE TAC 1995, Jiang et al., IEEE TAC 2001]

A is (Σ, k)-diagnosable if there is a k-diagnoser for A.

A is Σ-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is (Σ, k)-diagnosable.

Diagnosis Problems

- (A) Is A Σ-diagnosable ?
- (B) If "yes" to (A), compute the minimum k and
- (C) Compute a witness diagnoser.

Results for Diagnosis Problem

- (A) is in PTIME
- (B) is in PTIME
- (C) is in EXPTIME

A witness diagnoser is an automaton with at most $2^{O(A)}$ states

Results for the Diagnosis Problem

[Sampath et al., IEEE TAC 1995, Jiang et al., IEEE TAC 2001]

A is (Σ, k)-diagnosable if there is a k-diagnoser for A.

A is Σ-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is (Σ, k)-diagnosable.

Diagnosis Problems

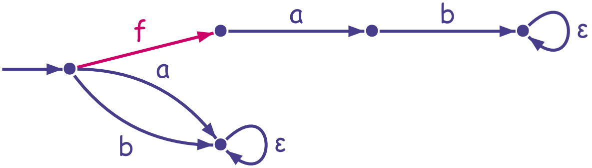
- (A) Is A Σ-diagnosable ?
- (B) If "yes" to (A), compute the minimum k and
- (C) Compute a witness diagnoser.

Results for Diagnosis Problem

- (A) is in PTIME
- (B) is in PTIME
- (C) is in EXPTIME

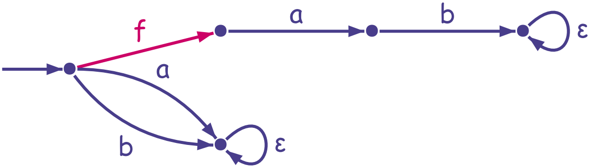
  A witness diagnoser is an automaton with at most $2^{O(A)}$ states

Minimization of the set of Observable Events



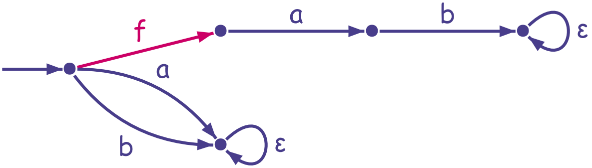
- Aim: find $\Sigma_o \subseteq \Sigma$ s.t. A is Σ_o-diagnosable and minimize $|\Sigma_o|$
- minimum $\Sigma_o = \{a, b\}$ and $(\Sigma_o, 2)$-diagnosable

Minimization of the set of Observable Events



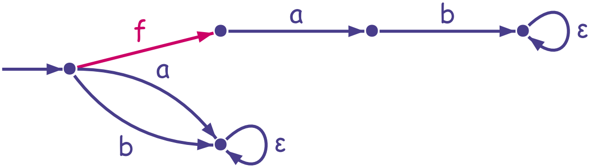
- Aim: find $\Sigma_o \subseteq \Sigma$ s.t. A is Σ_o-diagnosable and minimize $|\Sigma_o|$
- minimum $\Sigma_o = \{a, b\}$ and $(\Sigma_o, 2)$-diagnosable

Minimization of the set of Observable Events

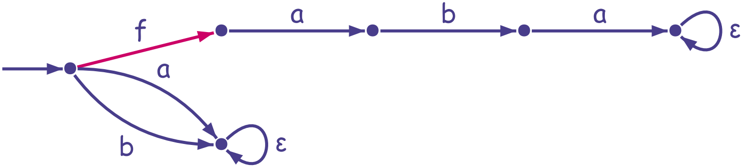


- Aim: find $\Sigma_o \subseteq \Sigma$ s.t. A is Σ_o-diagnosable and minimize $|\Sigma_o|$
- minimum $\Sigma_o = \{a, b\}$ and $(\Sigma_o, 2)$-diagnosable

Minimization of the set of Observable Events

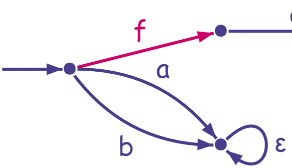


- Aim: find $\Sigma_o \subseteq \Sigma$ s.t. A is Σ_o-diagnosable and **minimize** $|\Sigma_o|$
- minimum $\Sigma_o = \{a, b\}$ and $(\Sigma_o, 2)$-diagnosable

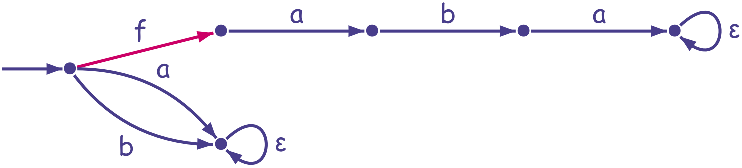


- $\Sigma_o = \{a\}$ is minimal and $(\Sigma_o, 3)$-diagnosable

Minimization of the set of Observable Events



- Aim: find $\Sigma_o \subseteq \Sigma$ s.t. A is Σ_o-diagnosable and minimize $|\Sigma_o|$
- minimum $\Sigma_o = \{a, b\}$ and $(\Sigma_o, 2)$-diagnosable



- $\Sigma_o = \{a\}$ is minimal and $(\Sigma_o, 3)$-diagnosable

Minimum Number of Observable Events

Problem 1: Minimum Number of Observable Events

Input: A, $n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any $\Sigma_o \subseteq \Sigma$, $|\Sigma_o| \leq n$, s.t. A is Σ_o-diagnosable ?

Theorem

Problem 1 is NP-complete.

- membership in NP: checking Σ-diagnosability is in P
- NP hardness: reduction of the n-clique problem to Problem 1.

Minimum Number of Observable Events

Problem 1: Minimum Number of Observable Events

Input: A, $n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any $\Sigma_o \subseteq \Sigma$, $|\Sigma_o| \leq n$, s.t. A is Σ_o-diagnosable ?

Theorem

Problem 1 is NP-complete.

- membership in NP: checking Σ-diagnosability is in P
- NP hardness: reduction of the n-clique problem to Problem 1.

Minimum Number of Observable Events

Problem 1: Minimum Number of Observable Events

Input: A, $n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any $\Sigma_o \subseteq \Sigma$, $|\Sigma_o| \leq n$, s.t. A is Σ_o-diagnosable ?

Theorem

Problem 1 is NP-complete.

- membership in NP: checking Σ-diagnosability is in P
- NP hardness: reduction of the n-clique problem to Problem 1.

Minimum Number of Observable Events

Problem 1: Minimum Number of Observable Events

Input: A, $n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any $\Sigma_o \subseteq \Sigma$, $|\Sigma_o| \leq n$, s.t. A is Σ_o-diagnosable ?

Theorem

Problem 1 is NP-complete.

- membership in NP: checking Σ-diagnosability is in P
- NP hardness: reduction of the n-clique problem to Problem 1.

Reduction of n-clique Problem to Problem 1

- $G = (V, E)$ an undirected graph
- $V' \subseteq V$ is a clique of G iff $\forall v, v' \in V', (v, v') \in E$
- n-clique: a clique V' with $|V'| = n$.

there is a n-clique in $G \iff A_G$ is $(\Sigma_o, 2)$-diagnosable, $|\Sigma \setminus \Sigma_o| = n$

Reduction of n-clique Problem to Problem 1

- $G = (V, E)$ an undirected graph
- $V' \subseteq V$ is a clique of G iff $\forall v, v' \in V', (v, v') \in E$
- n-clique: a clique V' with $|V'| = n$.

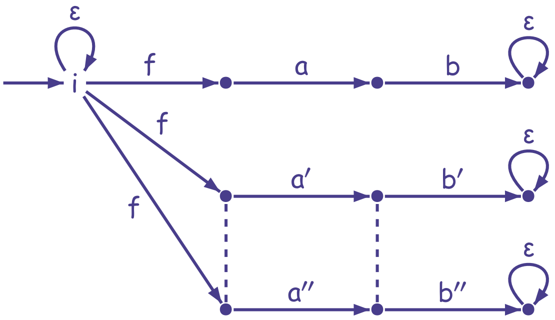
there is a n-clique in $G \iff A_G$ is $(\Sigma_o, 2)$-diagnosable, $|\Sigma \setminus \Sigma_o| = n$

Reduction of n-clique Problem to Problem 1

- $G = (V, E)$ an undirected graph
- $V' \subseteq V$ is a clique of G iff $\forall v, v' \in V', (v, v') \in E$
- n-clique: a clique V' with $|V'| = n$.

there is a n-clique in $G \iff A_G$ is $(\Sigma_o, 2)$-diagnosable, $|\Sigma \setminus \Sigma_o| = n$

Let $\Sigma = V$. For all $(a, b), (a', b') \cdots \notin E$: define A_G over $\Sigma^{\varepsilon, f}$ by



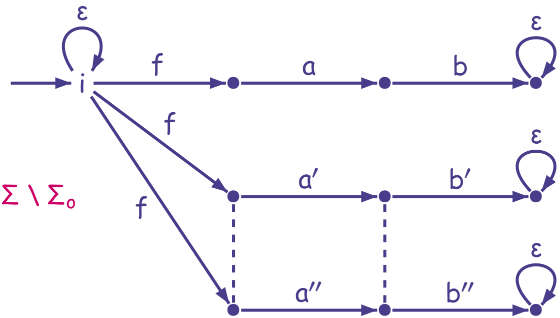
Reduction of n-clique Problem to Problem 1

- $G = (V, E)$ an undirected graph
- $V' \subseteq V$ is a clique of G iff $\forall v, v' \in V', (v, v') \in E$
- n-clique: a clique V' with $|V'| = n$.

there is a n-clique in G $\iff$ A_G is $(\Sigma_o, 2)$-diagnosable, $|\Sigma \setminus \Sigma_o| = n$

Let $\Sigma = V$. For all $(a, b), (a', b') \cdots \not\in E$: define A_G over $\Sigma^{\varepsilon, f}$ by

$\Longleftarrow$: $C = \Sigma \setminus \Sigma_o$



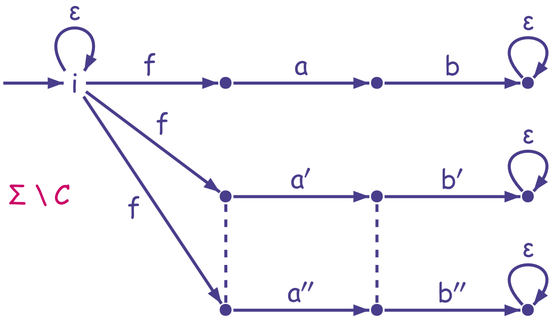
Reduction of n-clique Problem to Problem 1

- $G = (V, E)$ an undirected graph
- $V' \subseteq V$ is a clique of G iff $\forall v, v' \in V', (v, v') \in E$
- n-clique: a clique V' with $|V'| = n$.

there is a n-clique in $G \iff A_G$ is $(\Sigma_o, 2)$-diagnosable, $|\Sigma \setminus \Sigma_o| = n$

Let $\Sigma = V$. For all $(a, b), (a', b') \cdots \not\in E$: define A_G over $\Sigma^{\varepsilon, f}$ by

$\Longrightarrow$: $\Sigma_o = \Sigma \setminus C$



Variant: Renaming Observable Events

Definition (Mask)

A mask (M, n) over Σ is a mapping $M : \Sigma \longrightarrow \{1, \cdots, n\} \cup \{\varepsilon\}$.
M induces a mapping $M^* : \Sigma^* \longrightarrow \{1, \cdots, n\}^*$.

Definition (k-diagnoser for a Mask)

Let (M, n) be a mask over Σ. A mapping $D : \{1, \cdots, n\}^* \longrightarrow \{0, 1\}$ is a k-diagnoser for A if:

- for each $\rho \in$ **NonFaulty**(A), $D(M^*(tr(\rho))) = 0$;
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(M^*(tr(\rho))) = 1$.

A is $((M, n), k)$-diagnosable if there is a k-diagnoser.
A is (M, n)-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is $((M, n), k)$-diagnosable.

Mask Diagnosability: Given (M, n), is A (M, n)-diagnosable ?

Variant: Renaming Observable Events

Definition (Mask)

A mask (M, n) over Σ is a mapping $M : \Sigma \longrightarrow \{1, \cdots, n\} \cup \{\varepsilon\}$.
M induces a mapping $M^* : \Sigma^* \longrightarrow \{1, \cdots, n\}^*$.

Definition (k-diagnoser for a Mask)

Let (M, n) be a mask over Σ. A mapping $D : \{1, \cdots, n\}^* \longrightarrow \{0, 1\}$ is a k-diagnoser for A if:

- for each $\rho \in$ **NonFaulty**(A), $D(M^*(tr(\rho))) = 0$;
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(M^*(tr(\rho))) = 1$.

A is $((M, n), k)$-diagnosable if there is a k-diagnoser.
A is (M, n)-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is $((M, n), k)$-diagnosable.

Mask Diagnosability: Given (M, n), is A (M, n)-diagnosable ?

Variant: Renaming Observable Events

Definition (Mask)

A mask (M, n) over Σ is a mapping $M : \Sigma \longrightarrow \{1, \cdots, n\} \cup \{\varepsilon\}$.
M induces a mapping $M^* : \Sigma^* \longrightarrow \{1, \cdots, n\}^*$.

Definition (k-diagnoser for a Mask)

Let (M, n) be a mask over Σ. A mapping $D : \{1, \cdots, n\}^* \longrightarrow \{0, 1\}$ is a k-diagnoser for A if:

- for each $\rho \in$ **NonFaulty**(A), $D(M^*(tr(\rho))) = 0$;
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(M^*(tr(\rho))) = 1$.

A is $((M, n), k)$-diagnosable if there is a k-diagnoser.
A is (M, n)-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is $((M, n), k)$-diagnosable.

Mask Diagnosability: Given (M, n), is A (M, n)-diagnosable ?

Variant: Renaming Observable Events

Definition (Mask)

A mask (M, n) over Σ is a mapping $M : \Sigma \longrightarrow \{1, \cdots, n\} \cup \{\varepsilon\}$.
M induces a mapping $M^* : \Sigma^* \longrightarrow \{1, \cdots, n\}^*$.

Definition (k-diagnoser for a Mask)

Let (M, n) be a mask over Σ. A mapping $D : \{1, \cdots, n\}^* \longrightarrow \{0, 1\}$ is a k-diagnoser for A if:

- for each $\rho \in$ **NonFaulty**(A), $D(M^*(tr(\rho))) = 0$;
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(M^*(tr(\rho))) = 1$.

A is $((M, n), k)$-diagnosable if there is a k-diagnoser.
A is (M, n)-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is $((M, n), k)$-diagnosable.

Mask Diagnosability: Given (M, n), is A (M, n)-diagnosable ?

Variant: Renaming Observable Events

Definition (Mask)

A mask (M, n) over Σ is a mapping $M : \Sigma \longrightarrow \{1, \cdots, n\} \cup \{\varepsilon\}$.
M induces a mapping $M^* : \Sigma^* \longrightarrow \{1, \cdots, n\}^*$.

Definition (k-diagnoser for a Mask)

Let (M, n) be a mask over Σ. A mapping $D : \{1, \cdots, n\}^* \longrightarrow \{0, 1\}$ is a k-diagnoser for A if:

- for each $\rho \in \textbf{NonFaulty}(A)$, $D(M^*(tr(\rho))) = 0$;
- for each run $\rho \in \textbf{Faulty}_{\geq k}(A)$, $D(M^*(tr(\rho))) = 1$.

A is $((M, n), k)$-diagnosable if there is a k-diagnoser.
A is (M, n)-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is $((M, n), k)$-diagnosable.

Mask Diagnosability: Given (M, n), is A (M, n)-diagnosable ?

Variant: Renaming Observable Events

Definition (Mask)

A mask (M, n) over Σ is a mapping $M : \Sigma \longrightarrow \{1, \cdots, n\} \cup \{\varepsilon\}$.
M induces a mapping $M^* : \Sigma^* \longrightarrow \{1, \cdots, n\}^*$.

Definition (k-diagnoser for a Mask)

Let (M, n) be a mask over Σ. A mapping $D : \{1, \cdots, n\}^* \longrightarrow \{0, 1\}$ is a k-diagnoser for A if:

- for each $\rho \in$ **NonFaulty**(A), $D(M^*(tr(\rho))) = 0$;
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(M^*(tr(\rho))) = 1$.

A is $((M, n), k)$-diagnosable if there is a k-diagnoser.
A is (M, n)-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is $((M, n), k)$-diagnosable.

Mask Diagnosability: Given (M, n), is A (M, n)-diagnosable ?

Variant: Renaming Observable Events

Definition (Mask)

A mask (M, n) over Σ is a mapping $M : \Sigma \longrightarrow \{1, \cdots, n\} \cup \{\varepsilon\}$.
M induces a mapping $M^* : \Sigma^* \longrightarrow \{1, \cdots, n\}^*$.

Definition (k-diagnoser for a Mask)

Let (M, n) be a mask over Σ. A mapping $D : \{1, \cdots, n\}^* \longrightarrow \{0, 1\}$ is a k-diagnoser for A if:

- for each $\rho \in \mathbf{NonFaulty}(A)$, $D(M^*(tr(\rho))) = 0$;
- for each run $\rho \in \mathbf{Faulty}_{\geq k}(A)$, $D(M^*(tr(\rho))) = 1$.

A is $((M, n), k)$-diagnosable if there is a k-diagnoser.
A is (M, n)-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is $((M, n), k)$-diagnosable.

Mask Diagnosability: Given (M, n), is A (M, n)-diagnosable ?

Diagnosability Problem for Masks

Mask diagnosability

- given (M, n), rename every action $a \in \Sigma$ by $M(a)$
- obtain an a new automaton A_M with $\Sigma_M = \{1, \cdots, n\}$
- reduce to diagnosis problem: Is A_M Σ_M-diagnosable ?

Problem 2: Minimum Mask

Input: A, $n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

A solution to Problem 1 provides a solution to Problem 2.

- if A is Σ_o-diagnosable,
- let $M : \Sigma_o \rightarrow \{1, \cdots, |\Sigma_o|\}$, injective and $M(\Sigma \setminus \Sigma_o) = \varepsilon$
- then A is $(M, |\Sigma_o|)$-diagnosable.

Diagnosability Problem for Masks

Mask diagnosability

- given (M, n), rename every action $a \in \Sigma$ by $M(a)$
- obtain an a new automaton A_M with $\Sigma_M = \{1, \cdots, n\}$
- reduce to diagnosis problem: Is A_M Σ_M-diagnosable ?

Problem 2: Minimum Mask

Input: $A, n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

A solution to Problem 1 provides a solution to Problem 2.

- if A is Σ_o-diagnosable,
- let $M : \Sigma_o \rightarrow \{1, \cdots, |\Sigma_o|\}$, injective and $M(\Sigma \setminus \Sigma_o) = \varepsilon$
- then A is $(M, |\Sigma_o|)$-diagnosable.

Diagnosability Problem for Masks

Mask diagnosability

- given (M, n), rename every action $a \in \Sigma$ by $M(a)$
- obtain an a new automaton A_M with $\Sigma_M = \{1, \cdots, n\}$
- reduce to diagnosis problem: Is A_M Σ_M-diagnosable ?

Problem 2: Minimum Mask

Input: $A, n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

A solution to Problem 1 provides a solution to Problem 2.

- if A is Σ_o-diagnosable,
- let $M : \Sigma_o \to \{1, \cdots, |\Sigma_o|\}$, injective and $M(\Sigma \setminus \Sigma_o) = \varepsilon$
- then A is $(M, |\Sigma_o|)$-diagnosable.

Diagnosability Problem for Masks

Mask diagnosability

- given (M, n), rename every action $a \in \Sigma$ by $M(a)$
- obtain an a new automaton A_M with $\Sigma_M = \{1, \cdots, n\}$
- reduce to diagnosis problem: Is A_M Σ_M-diagnosable ?

Problem 2: Minimum Mask

Input: A, $n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

A solution to Problem 1 provides a solution to Problem 2.

- if A is Σ_o-diagnosable,
- let $M : \Sigma_o \to \{1, \cdots, |\Sigma_o|\}$, injective and $M(\Sigma \setminus \Sigma_o) = \varepsilon$
- then A is $(M, |\Sigma_o|)$-diagnosable.

Diagnosability Problem for Masks

Mask diagnosability

- given (M, n), rename every action $a \in \Sigma$ by $M(a)$
- obtain an a new automaton A_M with $\Sigma_M = \{1, \cdots, n\}$
- reduce to diagnosis problem: Is A_M Σ_M-diagnosable ?

Problem 2: Minimum Mask

Input: $A, n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

A solution to Problem 1 provides a solution to Problem 2.

- if A is Σ_o-diagnosable,
- let $M : \Sigma_o \to \{1, \cdots, |\Sigma_o|\}$, injective and $M(\Sigma \setminus \Sigma_o) = \varepsilon$
- then A is $(M, |\Sigma_o|)$-diagnosable.

Diagnosability Problem for Masks

Mask diagnosability

- given (M, n), rename every action $a \in \Sigma$ by $M(a)$
- obtain an a new automaton A_M with $\Sigma_M = \{1, \cdots, n\}$
- reduce to diagnosis problem: Is A_M Σ_M-diagnosable ?

Problem 2: Minimum Mask

Input: A, $n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

A solution to Problem 1 provides a solution to Problem 2.

- if A is Σ_o-diagnosable,
- let $M : \Sigma_o \to \{1, \cdots, |\Sigma_o|\}$, injective and $M(\Sigma \setminus \Sigma_o) = \varepsilon$
- then A is $(M, |\Sigma_o|)$-diagnosable.

Diagnosability Problem for Masks

Mask diagnosability

- given (M, n), rename every action $a \in \Sigma$ by $M(a)$
- obtain an a new automaton A_M with $\Sigma_M = \{1, \cdots, n\}$
- reduce to diagnosis problem: Is A_M Σ_M-diagnosable ?

Problem 2: Minimum Mask

Input: $A, n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

A solution to Problem 1 provides a solution to Problem 2.

- if A is Σ_o-diagnosable,
- let $M : \Sigma_o \to \{1, \cdots, |\Sigma_o|\}$, injective and $M(\Sigma \setminus \Sigma_o) = \varepsilon$
- then A is $(M, |\Sigma_o|)$-diagnosable.

Diagnosability Problem for Masks

Mask diagnosability

- given (M, n), rename every action $a \in \Sigma$ by $M(a)$
- obtain an a new automaton A_M with $\Sigma_M = \{1, \cdots, n\}$
- reduce to diagnosis problem: Is A_M Σ_M-diagnosable ?

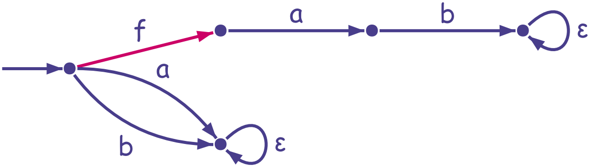
Problem 2: Minimum Mask

Input: $A, n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

A solution to Problem 1 provides a solution to Problem 2.

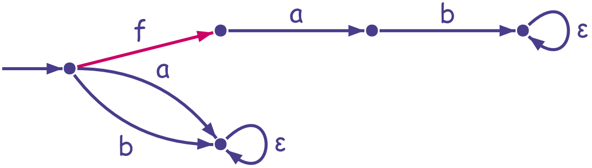
- if A is Σ_o-diagnosable,
- let $M : \Sigma_o \rightarrow \{1, \cdots, |\Sigma_o|\}$, injective and $M(\Sigma \setminus \Sigma_o) = \varepsilon$
- then A is $(M, |\Sigma_o|)$-diagnosable.

From Problem 2 to Problem 1 ?



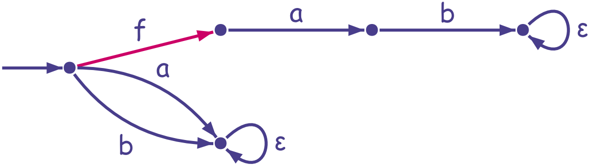
- $\mathcal{A}$ is $((M, 1), 2)$-diagnosable with $M(a) = M(b) = 1$.
  define $D(\varepsilon) = 0, D(1) = 0, D(1^2.\rho) = 1$.
- However $\mathcal{A}$ is not diagnosable for any $\Sigma_o \subseteq \Sigma$ s.t. $|\Sigma_o| = 1$.
  Neither $\Sigma_o = \{a\}$ nor $\Sigma_o = \{b\}$ can diagnose a fault.

From Problem 2 to Problem 1 ?



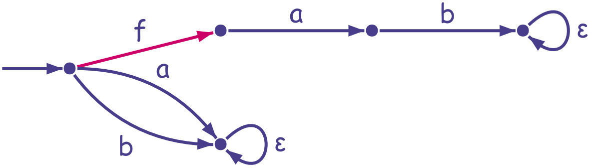
- $\mathcal{A}$ is ((M, 1), 2)-diagnosable with $M(a) = M(b) = 1$.
  define $D(\varepsilon) = 0, D(1) = 0, D(1^2.\rho) = 1$.
- However $\mathcal{A}$ is not diagnosable for any $\Sigma_o \subseteq \Sigma$ s.t. $|\Sigma_o| = 1$.
  Neither $\Sigma_o = \{a\}$ nor $\Sigma_o = \{b\}$ can diagnose a fault.

From Problem 2 to Problem 1 ?



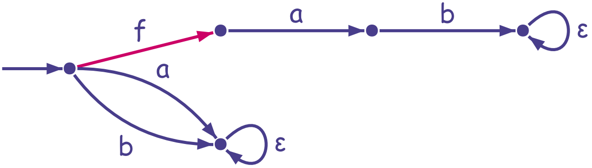
- $\mathcal{A}$ is $((M,1),2)$-diagnosable with $M(a) = M(b) = 1$.
  define $D(\varepsilon) = 0, D(1) = 0, D(1^2.\rho) = 1$.
- However $\mathcal{A}$ is not diagnosable for any $\Sigma_o \subseteq \Sigma$ s.t. $|\Sigma_o| = 1$.
  Neither $\Sigma_o = \{a\}$ nor $\Sigma_o = \{b\}$ can diagnose a fault.

From Problem 2 to Problem 1 ?



- $\mathcal{A}$ is $((M,1),2)$-diagnosable with $M(a) = M(b) = 1$.
  define $D(\varepsilon) = 0, D(1) = 0, D(1^2.\rho) = 1$.
- However $\mathcal{A}$ is not diagnosable for any $\Sigma_o \subseteq \Sigma$ s.t. $|\Sigma_o| = 1$.
  Neither $\Sigma_o = \{a\}$ nor $\Sigma_o = \{b\}$ can diagnose a fault.

From Problem 2 to Problem 1 ?



- $\mathcal{A}$ is ((M,1),2)-diagnosable with $M(a) = M(b) = 1$.
  define $D(\varepsilon) = 0, D(1) = 0, D(1^2.\rho) = 1$.
- However $\mathcal{A}$ is not diagnosable for any $\Sigma_o \subseteq \Sigma$ s.t. $|\Sigma_o| = 1$.
  Neither $\Sigma_o = \{a\}$ nor $\Sigma_o = \{b\}$ can diagnose a fault.

Minimum Mask

Problem 2: Minimum Mask

Input: $A, n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

Theorem

Problem 2 is NP-complete.

- membership in NP: checking (M, n)-diagnosability is in P
- NP hardness: reduction of the n-coloring problem to Problem 2.

Minimum Mask

Problem 2: Minimum Mask

Input: $A, n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

Theorem

Problem 2 is NP-complete.

- membership in NP: checking (M, n)-diagnosability is in P
- NP hardness: reduction of the n-coloring problem to Problem 2.

Minimum Mask

Problem 2: Minimum Mask

Input: $A, n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

Theorem

Problem 2 is NP-complete.

- membership in NP: checking (M, n)-diagnosability is in P
- NP hardness: reduction of the n-coloring problem to Problem 2.

Minimum Mask

Problem 2: Minimum Mask

Input: $A, n \in \mathbb{N}^*$ s.t. $n \leq |\Sigma|$.
Problem: Is there any mask (M, n) s.t. A is (M, n)-diagnosable ?

Theorem

Problem 2 is NP-complete.

- membership in NP: checking (M, n)-diagnosability is in P
- NP hardness: reduction of the n-coloring problem to Problem 2.

Reduction of n-Coloring Problem to Problem 2

- $G = (V, E)$ an undirected graph
- $C : V \longrightarrow \{1, \cdots, k\}$ is a proper coloring mapping if $(v, v') \in E$ implies $C(v) \neq C(v')$.
- n-coloring problem for G: is there any PCM $C : V \longrightarrow \{1, \cdots, n\}$?

G is n-colorizable $\iff$ A_G is $((M, n), 1)$-diagnosable

Reduction of n-Coloring Problem to Problem 2

- $G = (V, E)$ an undirected graph
- $C : V \longrightarrow \{1, \cdots, k\}$ is a proper coloring mapping if $(v, v') \in E$ implies $C(v) \neq C(v')$.
- n-coloring problem for G: is there any PCM $C : V \longrightarrow \{1, \cdots, n\}$?

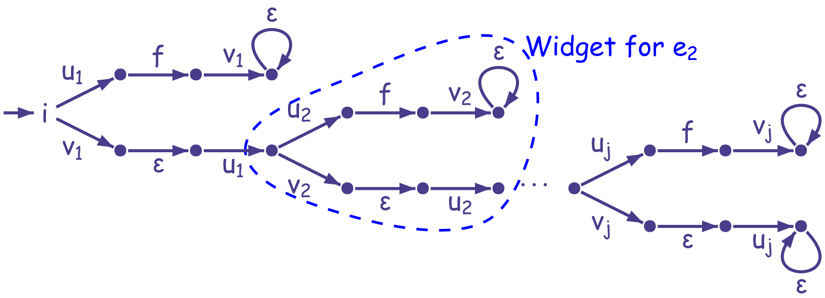
$$G \text{ is n-colorizable} \iff A_G \text{ is } ((M, n), 1)\text{-diagnosable}$$

Reduction of n-Coloring Problem to Problem 2

- $G = (V, E)$ an undirected graph
- $C : V \longrightarrow \{1, \cdots, k\}$ is a proper coloring mapping if $(v, v') \in E$ implies $C(v) \neq C(v')$.
- n-coloring problem for G: is there any PCM $C : V \longrightarrow \{1, \cdots, n\}$?

$$G \text{ is n-colorizable} \iff A_G \text{ is } ((M, n), 1)\text{-diagnosable}$$

Let $\Sigma = V$ and assume $E = \{e_1, \cdots, e_j\}$ with $e_j = (u_j, v_j)$.

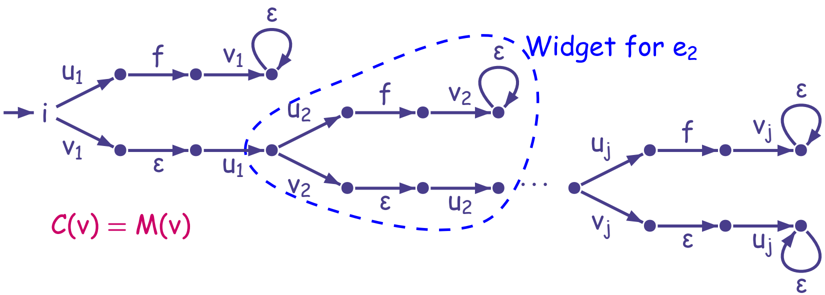


Reduction of n-Coloring Problem to Problem 2

- $G = (V, E)$ an undirected graph
- $C : V \longrightarrow \{1, \cdots, k\}$ is a proper coloring mapping if $(v, v') \in E$ implies $C(v) \neq C(v')$.
- n-coloring problem for G: is there any PCM $C : V \longrightarrow \{1, \cdots, n\}$?

$$G \text{ is n-colorizable} \iff A_G \text{ is } ((M, n), 1)\text{-diagnosable}$$

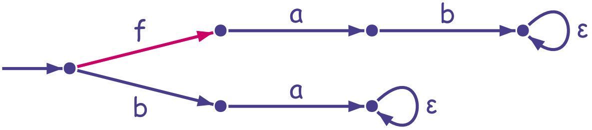
Let $\Sigma = V$ and assume $E = \{e_1, \cdots, e_j\}$ with $e_j = (u_j, v_j)$.



Outline

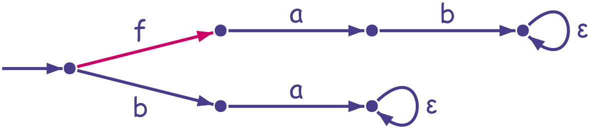
- Fault Diagnosis for Finite State Systems
- Fault Diagnosis with Dynamic Observers
  - Dynamic Observers
  - Diagnosability with Dynamic Observers
  - Synthesis of Dynamic Observers

Why Dynamic Observations ?



- Static observer: fixed set of observable events
- Static observation: $|\Sigma_o| \geq 2$, $\Sigma_o = \{a, b\}$; $(\{a, b\}, 1)$-diagnosable
- Dynamic observer: choose dynamically the set of observable events

Why Dynamic Observations ?

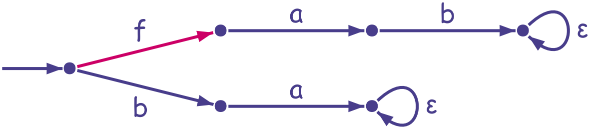


- Static observer: fixed set of observable events
- Static observation: $|\Sigma_o| \geq 2$, $\Sigma_o = \{a, b\}$; $(\{a, b\}, 1)$-diagnosable
- Dynamic observer: choose dynamically the set of observable events

Assume you can choose Σ_o dynamically:

> Is the plant diagnosable observing only one event at a time ?

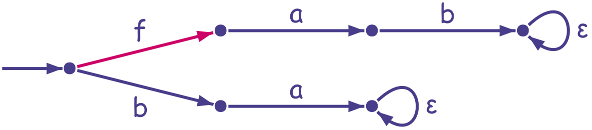
Why Dynamic Observations ?



- Static observer: fixed set of observable events
- Static observation: $|\Sigma_o| \geq 2$, $\Sigma_o = \{a, b\}$; $(\{a, b\}, 1)$-diagnosable
- Dynamic observer: choose dynamically the set of observable events

1. observe only a

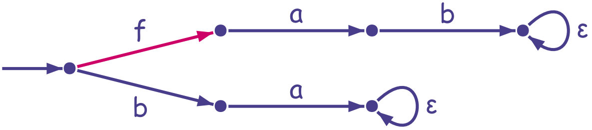
Why Dynamic Observations ?



- Static observer: fixed set of observable events
- Static observation: $|\Sigma_o| \geq 2$, $\Sigma_o = \{a, b\}$; $(\{a, b\}, 1)$-diagnosable
- Dynamic observer: choose dynamically the set of observable events

1. observe only a
2. once an a has been observed, observe only b

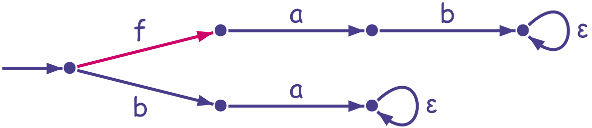
Why Dynamic Observations ?



- Static observer: fixed set of observable events
- Static observation: $|\Sigma_o| \geq 2$, $\Sigma_o = \{a, b\}$; $(\{a, b\}, 1)$-diagnosable
- Dynamic observer: choose dynamically the set of observable events

1. observe only a
2. once an a has been observed, observe only b
3. if a.b occurs diagnose a fault

Why Dynamic Observations ?



- Static observer: fixed set of observable events
- Static observation: $|\Sigma_o| \geq 2$, $\Sigma_o = \{a, b\}$; $(\{a, b\}, 1)$-diagnosable
- Dynamic observer: choose dynamically the set of observable events

1. observe only a
2. once an a has been observed, observe only b
3. if a.b occurs diagnose a fault

The plant is "dynamically" 2-diagnosable

Dynamic Observers

Definition (Dynamic Observer)

A dynamic observer Obs is a complete and deterministic labeled automaton $(S, s_0, \Sigma, \delta, L)$ s.t. $\forall s \in S$, $\forall a \in \Sigma$, if $a \notin L(s)$ then $\delta(s, a) = s$.

Definition ((Obs, k)-diagnoser)

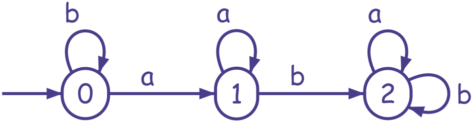
$D : \Sigma^* \longrightarrow \{0, 1\}$ is an (Obs, k)-diagnoser for A if

- for each run $\rho \in$ **NonFaulty**(A), $D(Obs(\pi_{/\Sigma}(tr(\rho)))) = 0$ and
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(Obs(\pi_{/\Sigma}(tr(\rho)))) = 1$.

Dynamic Observers

Definition (Dynamic Observer)

A dynamic observer Obs is a complete and deterministic labeled automaton $(S, s_0, \Sigma, \delta, L)$ s.t. $\forall s \in S$, $\forall a \in \Sigma$, if $a \notin L(s)$ then $\delta(s, a) = s$.



$L(0) = \{a\}$ $L(1) = \{b\}$ $L(2) = \emptyset$

Definition ((Obs, k)-diagnoser)

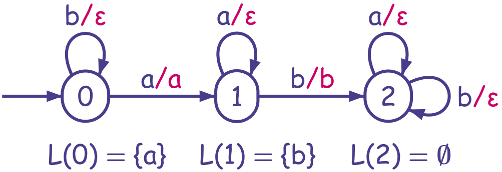
$D : \Sigma^* \longrightarrow \{0, 1\}$ is an (Obs, k)-diagnoser for A if

- for each run $\rho \in$ **NonFaulty**(A), $D(Obs(\pi_{/\Sigma}(tr(\rho)))) = 0$ and
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(Obs(\pi_{/\Sigma}(tr(\rho)))) = 1$.

Dynamic Observers

Definition (Dynamic Observer)

A dynamic observer Obs is a complete and deterministic labeled automaton $(S, s_0, \Sigma, \delta, L)$ s.t. $\forall s \in S$, $\forall a \in \Sigma$, if $a \notin L(s)$ then $\delta(s, a) = s$.

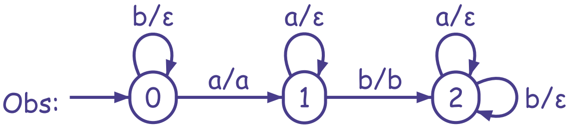


Definition ((Obs, k)-diagnoser)

$D : \Sigma^* \longrightarrow \{0, 1\}$ is an (Obs, k)-diagnoser for A if

- for each run $\rho \in$ **NonFaulty**(A), $D(Obs(\pi_{/\Sigma}(tr(\rho)))) = 0$ and
- for each run $\rho \in$ **Faulty**$_{\geq k}(A)$, $D(Obs(\pi_{/\Sigma}(tr(\rho)))) = 1$.

Dynamic Observer and Diagnosability

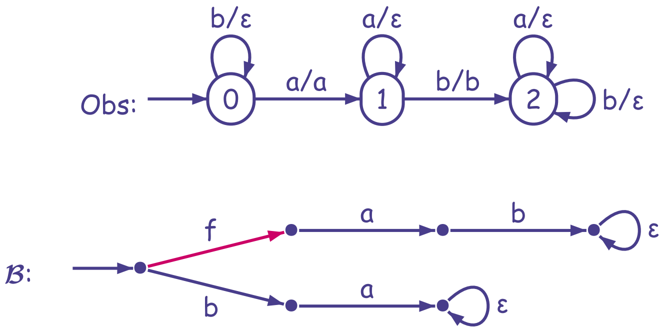


$\mathcal{B}$ is (Obs, 2)-diagnosable.

Define $D(a.b.\rho) = 1$ and $D(\rho) = 0$ otherwise. D is 2-diagnoser.

Obs observes only one event in each state.

Dynamic Observer and Diagnosability

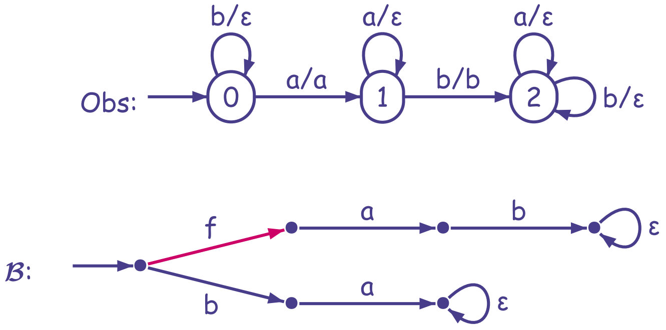


$\mathcal{B}$ is $(\mathrm{Obs}, 2)$-diagnosable.

Define $D(a.b.\rho) = 1$ and $D(\rho) = 0$ otherwise. D is 2-diagnoser.

Obs observes only one event in each state.

Dynamic Observer and Diagnosability

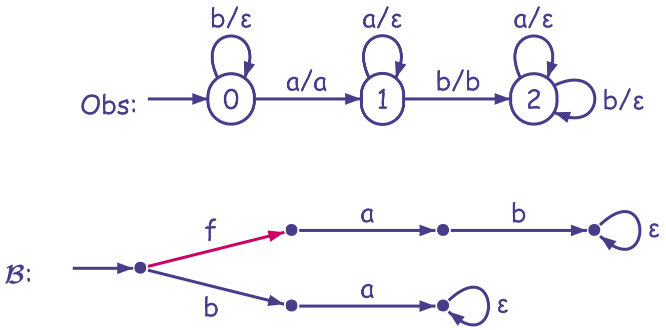


$\mathcal{B}$ is (Obs, 2)-diagnosable.

Define $D(a.b.\rho) = 1$ and $D(\rho) = 0$ otherwise. D is 2-diagnoser.

Obs observes only one event in each state.

Dynamic Observer and Diagnosability



$\mathcal{B}$ is (Obs, 2)-diagnosable.

Define $D(a.b.\rho) = 1$ and $D(\rho) = 0$ otherwise. D is 2-diagnoser.

Obs observes only one event in each state.

Checking Obs-diagnosability

(Obs, k)-diagnosability: A is (Obs, k)-diagnosable if there is an (Obs, k)-diagnoser for A

Obs-diagnosability: A is Obs-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is (Obs, k)-diagnosable

Problem 3: Finite Obs-diagnosability Problem

Input: A, a finite state observer Obs.
Problem: Is A Obs-diagnosable ?

To check Obs-diagnosability, build a product $A \otimes Obs$:

- initial state: (q_0, s_0)
- $(q,s) \xrightarrow{\beta} (q',s')$ iff $q \xrightarrow{\lambda} q'$, $s \xrightarrow{\lambda/\beta} s'$ for $\lambda \in \Sigma$,
- $(q,s) \xrightarrow{\lambda} (q',s)$ iff $q \xrightarrow{\lambda} q'$, and $\lambda \in \{\varepsilon, f\}$

A is Obs-diagnosable iff $A \otimes Obs$ is Σ-diagnosable

Checking Obs-diagnosability

(Obs, k)-diagnosability: A is (Obs, k)-diagnosable if there is an (Obs, k)-diagnoser for A

Obs-diagnosability: A is Obs-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is (Obs, k)-diagnosable

Problem 3: Finite Obs-diagnosability Problem

Input: A, a finite state observer Obs.
Problem: Is A Obs-diagnosable ?

To check Obs-diagnosability, build a product $A \otimes Obs$:

- initial state: (q_0, s_0)
- $(q, s) \xrightarrow{\beta} (q', s')$ iff $q \xrightarrow{\lambda} q'$, $s \xrightarrow{\lambda/\beta} s'$ for $\lambda \in \Sigma$,
- $(q, s) \xrightarrow{\lambda} (q', s)$ iff $q \xrightarrow{\lambda} q'$, and $\lambda \in \{\varepsilon, f\}$

A is Obs-diagnosable iff $A \otimes Obs$ is Σ-diagnosable

Checking Obs-diagnosability

(Obs, k)-diagnosability: A is (Obs, k)-diagnosable if there is an (Obs, k)-diagnoser for A

Obs-diagnosability: A is Obs-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is (Obs, k)-diagnosable

Problem 3: Finite Obs-diagnosability Problem

Input: A, a finite state observer Obs.
Problem: Is A Obs-diagnosable ?

To check Obs-diagnosability, build a product $A \otimes Obs$:

- initial state: (q_0, s_0)
- $(q,s) \xrightarrow{\beta} (q',s')$ iff $q \xrightarrow{\lambda} q'$, $s \xrightarrow{\lambda/\beta} s'$ for $\lambda \in \Sigma$,
- $(q,s) \xrightarrow{\lambda} (q',s)$ iff $q \xrightarrow{\lambda} q'$, and $\lambda \in \{\varepsilon, f\}$

A is Obs-diagnosable iff $A \otimes Obs$ is Σ-diagnosable

Checking Obs-diagnosability

(Obs, k)-diagnosability: A is (Obs, k)-diagnosable if there is an (Obs, k)-diagnoser for A

Obs-diagnosability: A is Obs-diagnosable if $\exists k \in \mathbb{N}$ s.t. A is (Obs, k)-diagnosable

Problem 3: Finite Obs-diagnosability Problem

Input: A, a finite state observer Obs.
Problem: Is A Obs-diagnosable ?

To check Obs-diagnosability, build a product $A \otimes \text{Obs}$:

- initial state: (q_0, s_0)
- $(q, s) \xrightarrow{\beta} (q', s')$ iff $q \xrightarrow{\lambda} q'$, $s \xrightarrow{\lambda/\beta} s'$ for $\lambda \in \Sigma$,
- $(q, s) \xrightarrow{\lambda} (q', s)$ iff $q \xrightarrow{\lambda} q'$, and $\lambda \in \{\varepsilon, f\}$

A is Obs-diagnosable iff $A \otimes \text{Obs}$ is Σ-diagnosable

Most Permissive Observer

Problem 4: Dynamic Diagnosability Problem

Input: A.
Problem: Compute the set of all observers s.t. A is Obs-diagnosable.

Too Hard ...

Problem 5: Dynamic k-Diagnosability Problem

Input: A, $k \in \mathbb{N}$.
Problem: Compute the set of all observers s.t. A is (Obs, k)-diagnosable.

Most Permissive Observer

Problem 4: Dynamic Diagnosability Problem

Input: A.
Problem: Compute the set of all observers s.t. A is Obs-diagnosable.

Too Hard ...

Problem 5: Dynamic k-Diagnosability Problem

Input: A, $k \in \mathbb{N}$.
Problem: Compute the set of all observers s.t. A is (Obs, k)-diagnosable.

Most Permissive Observer

Problem 4: Dynamic Diagnosability Problem

Input: A.
Problem: Compute the set of all observers s.t. A is Obs-diagnosable.

Too Hard ...

Problem 5: Dynamic k-Diagnosability Problem

Input: $A, k \in \mathbb{N}$.
Problem: Compute the set of all observers s.t. A is (Obs, k)-diagnosable.

Problem 5 as a Game Problem

- Reduce Problem 5 to a safety 2-player game
- Player 1 chooses what to observe
- Player 2 tries to produce a bad sequence of events
  i.e. traces that are in **NonFaulty**$(A) \cap$ **Faulty**$_{\geq k}(A)$
  behaves like a synchronized product of two automata

Problem 5 as a Game Problem

- Reduce Problem 5 to a safety 2-player game
- Player 1 chooses what to observe
- Player 2 tries to produce a bad sequence of events
  i.e. traces that are in $\mathbf{NonFaulty}(A) \cap \mathbf{Faulty}_{\geq k}(A)$
  behaves like a synchronized product of two automata

Problem 5 as a Game Problem

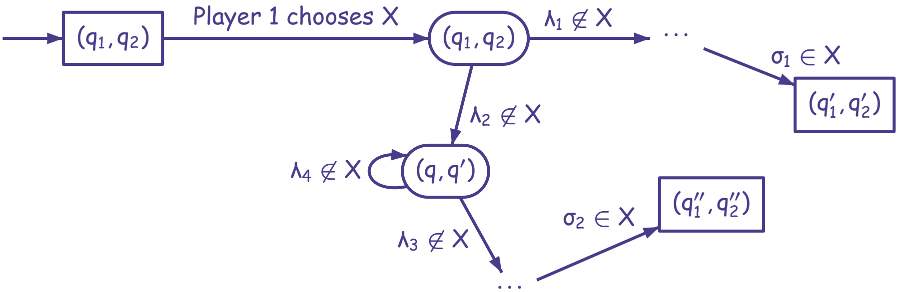
- Reduce Problem 5 to a safety 2-player game
- Player 1 chooses what to observe
- Player 2 tries to produce a bad sequence of events
  i.e. traces that are in $\textbf{NonFaulty}(A) \cap \textbf{Faulty}_{\geq k}(A)$
  behaves like a synchronized product of two automata

Problem 5 as a Game Problem

- Reduce Problem 5 to a safety 2-player game
- Player 1 chooses what to observe
- Player 2 tries to produce a bad sequence of events
  i.e. traces that are in **NonFaulty**$(A) \cap$ **Faulty**$_{\geq k}(A)$
  behaves like a synchronized product of two automata

Problem 5 as a Game Problem

- Reduce Problem 5 to a safety 2-player game
- Player 1 chooses what to observe
- Player 2 tries to produce a bad sequence of events
  i.e. traces that are in $\textbf{NonFaulty}(A) \cap \textbf{Faulty}_{\geq k}(A)$
  behaves like a synchronized product of two automata



Results:

Theorem

O an observer s.t. A is (O, k)-diagnosable $\implies$ there is a corresponding winning strategy in G.

Theorem

There is a winning strategy in G $\implies$ there is a corresponding observer O s.t. A is (O, k)-diagnosable.

Theorem (Memoryless Strategies are Sufficient)

There is a memoryless most permissive strategy for any (safety) finite game G.

Theorem (Existence of a Most Permissive Observer)

There is a finite-state (2EXP) most permissive observer.

Results:

Theorem

O an observer s.t. A is (O, k)-diagnosable $\implies$ there is a corresponding winning strategy in G.

Theorem

There is a winning strategy in G $\implies$ there is a corresponding observer O s.t. A is (O, k)-diagnosable.

Theorem (Memoryless Strategies are Sufficient)

There is a memoryless most permissive strategy for any (safety) finite game G.

Theorem (Existence of a Most Permissive Observer)

There is a finite-state (2EXP) most permissive observer.

Results:

Theorem

O an observer s.t. A is (O, k)-diagnosable $\implies$ there is a corresponding winning strategy in G.

Theorem

There is a winning strategy in G $\implies$ there is a corresponding observer O s.t. A is (O, k)-diagnosable.

Theorem (Memoryless Strategies are Sufficient)

There is a memoryless most permissive strategy for any (safety) finite game G.

Theorem (Existence of a Most Permissive Observer)

There is a finite-state (2EXP) most permissive observer.

Results:

Theorem

O an observer s.t. A is (O, k)-diagnosable $\implies$ there is a corresponding winning strategy in G.

Theorem

There is a winning strategy in G $\implies$ there is a corresponding observer O s.t. A is (O, k)-diagnosable.

Theorem (Memoryless Strategies are Sufficient)

There is a memoryless most permissive strategy for any (safety) finite game G.

Theorem (Existence of a Most Permissive Observer)

There is a finite-state (2EXP) most permissive observer.

Results:

Theorem

O an observer s.t. A is (O,k)-diagnosable $\Longrightarrow$ there is a corresponding winning strategy in G.

Theorem

There is a winning strategy in G $\Longrightarrow$ there is a corresponding observer O s.t. A is (O,k)-diagnosable.

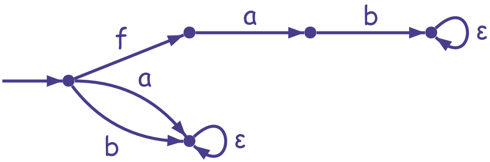
Theorem (Memoryless Strategies are Sufficient)

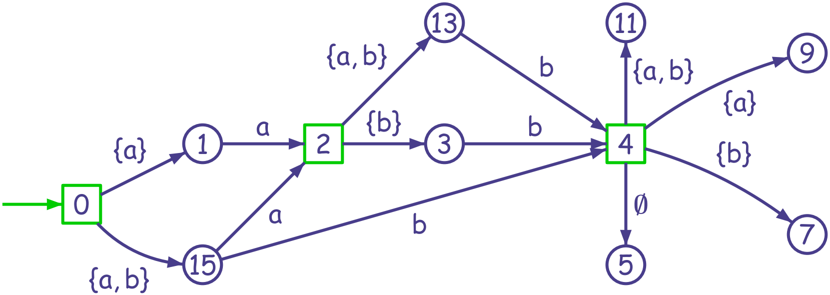
There is a memoryless most permissive strategy for any (safety) finite game G.

Theorem (Existence of a Most Permissive Observer)

There is a finite-state (2EXP) most permissive observer.

Example: Most Permissive Observer





Conclusion & Future Work

Results:

- Dynamic observers for bounded diagnosis (k-diagnosability)
- Computation of the most permissive observer
  [C. Tripakis Altisen, ACSD 2007]
- Cost & computation of the cost of a dynamic observers
- Existence of a finite optimal dynamic observer
- Effective computation of the optimal dynamic observer
  [C. Tripakis Altisen, TASE 2007]

Future Work:

- Exact complexity of Problem 2
- Implement the algorithm
- Extend to control of DES

Conclusion & Future Work

Results:

- Dynamic observers for bounded diagnosis (k-diagnosability)
- Computation of the most permissive observer
  [C. Tripakis Altisen, ACSD 2007]
- Cost & computation of the cost of a dynamic observers
- Existence of a finite optimal dynamic observer
- Effective computation of the optimal dynamic observer
  [C. Tripakis Altisen, TASE 2007]

Future Work:

- Exact complexity of Problem 2
- Implement the algorithm
- Extend to control of DES

Conclusion & Future Work

Results:

- Dynamic observers for bounded diagnosis (k-diagnosability)
- Computation of the most permissive observer
  [C. Tripakis Altisen, ACSD 2007]
- Cost & computation of the cost of a dynamic observers
- Existence of a finite optimal dynamic observer
- Effective computation of the optimal dynamic observer
  [C. Tripakis Altisen, TASE 2007]

Future Work:

- Exact complexity of Problem 2
- Implement the algorithm
- Extend to control of DES

References

[C. Tripakis Altisen, 2007] Franck Cassez, Stavros Tripakis, and Karine Altisen.
Sensor minimization problems with static or dynamic observers for fault diagnosis.
Technical Report RI-2007-1, IRCCyN/CNRS, 1 rue de la Noë, BP 92101, 44321 Nantes Cedes, France, January 2007.
Available at http://www.irccyn.fr/franck.

[C. Tripakis Altisen, ACSD 2007] Franck Cassez, Stavros Tripakis, and Karine Altisen.
Sensor minimization problems with static or dynamic observers for fault diagnosis.
In *7th International Conference on Application of Concurrency to System Design (ACSD'07)*, Bratislava, Slovak Republic, July 2007. IEEE Computer Society.

[C. Tripakis Altisen, TASE 2007] Franck Cassez, Stavros Tripakis, and Karine Altisen.
Synthesis Of Optimal-Cost Dynamic Observers for Fault Diagnosis of Discrete-Event Systems
In *1st IFIP & IEEE International Conference on Theoretical Aspects of Software Engineering (TASE'07)*, Shanghai, China, June 2007. IEEE Computer Society.

[Dasdan et al., IEEE DAC 1999] Ali Dasdan, Sandy S. Irani, and Rajesh K. Gupta.
Efficient algorithms for optimum cycle mean and optimum cost to time ratio problems.
In *Annual ACM IEEE Design Automation Conference*, pages 37–42, New Orleans, Louisiana, United States, 1999. ACM Press New York, NY, USA.
ISBN:1-58133-109-7.

[Debouk et al., DEDS 2004] Rami Debouk, Stéphane Lafortune, and Demosthenis Teneketzis.
On an optimization problem in sensor selection.
Discrete Event Dynamic Systems, 4(12), November 2004.

[Jiang et al., IEEE TAC 2001] Shengbing Jiang, Zhongdong Huang, Vigyan Chandra, and Ratnesh Kumar.
A polynomial algorithm for testing diagnosability of discrete event systems.
IEEE Transactions on Automatic Control, 46(8), August 2001.

[Jiang et al., IEEE TAC 2003] Shengbing Jiang, Ratnesh Kumar, and Humberto E. Garcia.
Optimal sensor selection for discrete event systems with partial observation.
IEEE Transactions on Automatic Control, 48(3):369–381, March 2003.

[Karp, Discrete Math. 1978] Richard M. Karp.
A characterization of the minimum mean cycle in a digraph.
Discrete Mathematics, 23:309–311, 1978.

References (cont.)

[R & W, 1987] Peter Ramadge and W. Murray Wonham.
Supervisory control of a class of discrete event processes.
SIAM J. Control Optim., 25(1), January 1987.

[Sampath et al., IEEE TAC 1995] Meera Sampath, Raja Sengupta, Stephane Lafortune, Kasim Sinnamohideen, and Demosthenis C. Teneketzis.
Diagnosability of discrete event systems.
IEEE Transactions on Automatic Control, 40(9), September 1995.

[Tsitsiklis, 1989] John N. Tsitsiklis.
On the control of discrete event dynamical systems.
Mathematics of Control, Signals and Systems, 2(2), 1989.

[Yoo et al., ACC 2001] Tae-Sic Yoo and Stéphane Lafortune.
On the computational complexity of some problems arising in partially-observed discrete event systems.
In *American Control Conference (ACC'01)*, 2001.
Arlington, VA.

[Zwick & Paterson, TCS 1996] Uri Zwick and Michael S. Paterson.
The complexity of mean payoff games on graphs.
Theoretical Computer Science, 158(1–2):343–359, 1996.

Algorithm for Checking Diagnosability

Necessary and Sufficient Condition for Diagnosability:

$$A \text{ is not } \Sigma\text{-diagnosable} \iff \forall k \in \mathbb{N}^*, A \text{ is not } (\Sigma, k)\text{-diagnosable}$$

$$\iff \forall k \in \mathbb{N}^*, \begin{cases} \exists \rho \in \textbf{NonFaulty}(A) \\ \exists \rho' \in \textbf{Faulty}_{\geq k}(A) \\ \qquad \pi_{/\Sigma}(\rho) = \pi_{/\Sigma}(\rho') \end{cases}$$

Algorithm for Checking Diagnosability

Necessary and Sufficient Condition for Diagnosability:

$$A \text{ is not } \Sigma\text{-diagnosable} \iff \forall k \in \mathbb{N}^*, A \text{ is not } (\Sigma, k)\text{-diagnosable}$$

$$\iff \forall k \in \mathbb{N}^*, \begin{cases} \exists \rho \in \textbf{NonFaulty}(A) \\ \exists \rho' \in \textbf{Faulty}_{\geq k}(A) \\ \qquad \pi_{/\Sigma}(\rho) = \pi_{/\Sigma}(\rho') \end{cases}$$

Let $A_1 = (Q \times \{0,1\}, (q_0, 0), \Sigma^{\varepsilon_1}, \longrightarrow_1)$ s.t.

- $(q,k) \xrightarrow{l}_1 (q',k')$ iff $q \xrightarrow{l} q'$ and $l \in \Sigma$ and $k = k'$;
- $(q,k) \xrightarrow{\varepsilon_1}_1 (q',1)$ iff $q \xrightarrow{f} q'$
  k is set to 1 after a fault occurs and will remain 1
- $(q,k) \xrightarrow{\varepsilon_1}_1 (q',k)$ iff $q \xrightarrow{\varepsilon} q'$.

Algorithm for Checking Diagnosability

Necessary and Sufficient Condition for Diagnosability:

$$A \text{ is not } \Sigma\text{-diagnosable} \iff \forall k \in \mathbb{N}^*, A \text{ is not } (\Sigma, k)\text{-diagnosable}$$

$$\iff \forall k \in \mathbb{N}^*, \begin{cases} \exists \rho \in \textbf{NonFaulty}(A) \\ \exists \rho' \in \textbf{Faulty}_{\geq k}(A) \\ \qquad \pi_{/\Sigma}(\rho) = \pi_{/\Sigma}(\rho') \end{cases}$$

Define $A_2 = (Q, q_0, \Sigma^{\varepsilon_2}, \longrightarrow_2)$ with

- $q \xrightarrow{l}_2 q'$ if $q \xrightarrow{l} q'$ and $l \in \Sigma$;
- $q \xrightarrow{\varepsilon_2}_2 q'$ if $q \xrightarrow{\varepsilon} q'$.

Algorithm for Checking Diagnosability

Necessary and Sufficient Condition for Diagnosability:

$$A \text{ is not } \Sigma\text{-diagnosable} \iff \forall k \in \mathbb{N}^*, A \text{ is not } (\Sigma, k)\text{-diagnosable}$$

$$\iff \forall k \in \mathbb{N}^*, \begin{cases} \exists \rho \in \textbf{NonFaulty}(A) \\ \exists \rho' \in \textbf{Faulty}_{\geq k}(A) \\ \qquad \pi_{/\Sigma}(\rho) = \pi_{/\Sigma}(\rho') \end{cases}$$

Let $\mathcal{B} = A_1 \times A_2$

Büchi acceptance condition: infinitely many faulty states and A_1-actions

Algorithm for Checking Diagnosability

Necessary and Sufficient Condition for Diagnosability:

$$A \text{ is not } \Sigma\text{-diagnosable} \iff \forall k \in \mathbb{N}^*, A \text{ is not } (\Sigma, k)\text{-diagnosable}$$

$$\iff \forall k \in \mathbb{N}^*, \begin{cases} \exists \rho \in \textbf{NonFaulty}(A) \\ \exists \rho' \in \textbf{Faulty}_{\geq k}(A) \\ \qquad \pi_{/\Sigma}(\rho) = \pi_{/\Sigma}(\rho') \end{cases}$$

Let $\mathcal{B} = A_1 \times A_2$

Büchi acceptance condition: infinitely many faulty states and A_1-actions

Theorem

$\mathcal{L}^\omega(\mathcal{B}) \neq \emptyset \iff$ *A is not Σ-diagnosable.*

Algorithm for Checking Diagnosability

Necessary and Sufficient Condition for Diagnosability:

$$A \text{ is not } \Sigma\text{-diagnosable} \iff \forall k \in \mathbb{N}^*, A \text{ is not } (\Sigma, k)\text{-diagnosable}$$

$$\iff \forall k \in \mathbb{N}^*, \begin{cases} \exists \rho \in \textbf{NonFaulty}(A) \\ \exists \rho' \in \textbf{Faulty}_{\geq k}(A) \\ \qquad \pi_{/\Sigma}(\rho) = \pi_{/\Sigma}(\rho') \end{cases}$$

Let $\mathcal{B} = A_1 \times A_2$

Büchi acceptance condition: infinitely many faulty states and A_1-actions

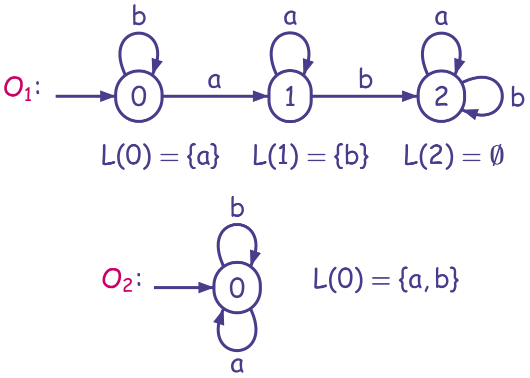
Theorem

$\mathcal{L}^\omega(\mathcal{B}) \neq \emptyset \iff A$ is not Σ-diagnosable.

Theorem

The minimum k s.t. A is (Σ, k)-diagnosable can be computed in PTIME.

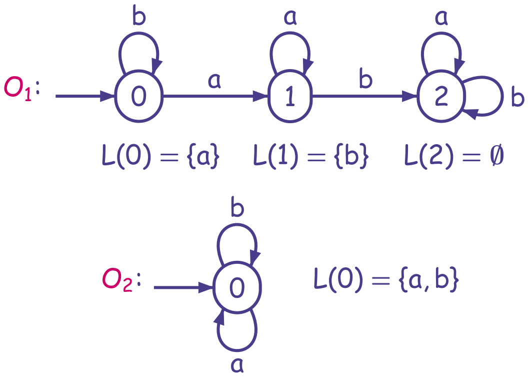
Comparing Dynamic Observers



- O_1 observes less events than O_2 in the long run
- O_1 is less expensive than O_2

New Problem: compute an optimal observer

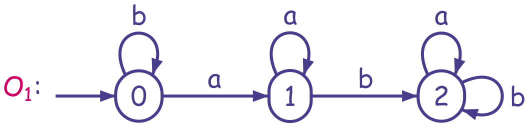
Comparing Dynamic Observers



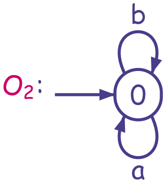
- O_1 observes less events than O_2 in the long run
- O_1 is less expensive than O_2

New Problem: compute an optimal observer

Comparing Dynamic Observers



$L(0) = \{a\}$ $L(1) = \{b\}$ $L(2) = \emptyset$



$L(0) = \{a, b\}$

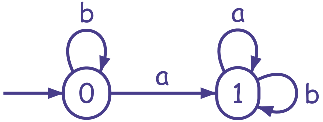
- O_1 observes less events than O_2 in the long run
- O_1 is less expensive than O_2

New Problem: compute an optimal observer

Cost of an Observer

Cost of a run: average number of events observed along a run

Let $Obs = (S, s_0, \Sigma, \delta, L)$



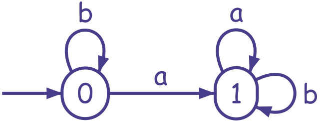
$L(0) = \{a\}$
$L(1) = \emptyset$

- cost relative to observation of the output w of the plant:

- cost relative to the raw output w of the plant:

Cost of an Observer

Cost of a run: average number of events observed along a run
Let $Obs = (S, s_0, \Sigma, \delta, L)$



$L(0) = \{a\}$
$L(1) = \emptyset$

- cost relative to observation of the output w of the plant:

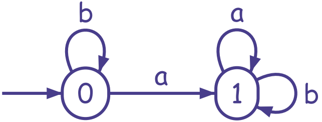
$$\mathbf{Cost}_1(w) = \frac{\Sigma_{i=0}^{i=n} |L(\delta(s_0, Obs(w)(i)))|}{n+1} \quad \textit{with } n = |Obs(w)|$$

- cost relative to the raw output w of the plant:

Cost of an Observer

Cost of a run: average number of events observed along a run

Let $Obs = (S, s_0, \Sigma, \delta, L)$



$L(0) = \{a\}$

$L(1) = \emptyset$

- cost relative to observation of the output w of the plant:

$$\mathbf{Cost}_1(w) = \frac{\Sigma_{i=0}^{i=n} |L(\delta(s_0, Obs(w)(i)))|}{n+1} \quad \textit{with } n = |Obs(w)|$$

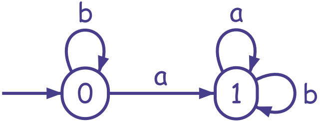
$Obs(b^n.a) = a$ and $\mathbf{Cost}_1(b^n.a) = 1/2$

- cost relative to the raw output w of the plant:

Cost of an Observer

Cost of a run: average number of events observed along a run

Let $Obs = (S, s_0, \Sigma, \delta, L)$



$L(0) = \{a\}$
$L(1) = \emptyset$

- cost relative to observation of the output w of the plant:

$$\mathbf{Cost}_1(w) = \frac{\Sigma_{i=0}^{i=n} |L(\delta(s_0, Obs(w)(i)))|}{n+1} \quad \textit{with } n = |Obs(w)|$$

$Obs(b^n.a) = a$ and $\mathbf{Cost}_1(b^n.a) = 1/2$

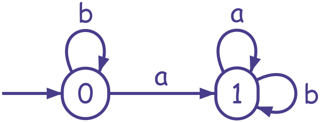
- cost relative to the raw output w of the plant:

$$\mathbf{Cost}_2(w) = \frac{\Sigma_{i=0}^{i=n} |L(\delta(s_0, w(i)))|}{n+1} \quad \textit{with } n = |w|$$

Cost of an Observer

Cost of a run: average number of events observed along a run

Let $Obs = (S, s_0, \Sigma, \delta, L)$



$L(0) = \{a\}$
$L(1) = \emptyset$

- cost relative to observation of the output w of the plant:

$$\mathbf{Cost}_1(w) = \frac{\Sigma_{i=0}^{i=n} |L(\delta(s_0, Obs(w)(i)))|}{n+1} \quad \textit{with } n = |Obs(w)|$$

$Obs(b^n.a) = a$ and $\mathbf{Cost}_1(b^n.a) = 1/2$

- cost relative to the raw output w of the plant:

$$\mathbf{Cost}_2(w) = \frac{\Sigma_{i=0}^{i=n} |L(\delta(s_0, w(i)))|}{n+1} \quad \textit{with } n = |w|$$

$\mathbf{Cost}_2(b^n.a) = (n+1)/(n+2)$

Computing the Cost of a Given Observer

$$\rho = q_0 \xrightarrow{a_1} q_1 \cdots q_{n-1} \xrightarrow{a_n} q_n \text{ a run of the plant } A$$

Let Obs be an observer and $w_i = \pi_{/\Sigma}(tr(q_0 \cdots q_i))$

$$\mathbf{Cost}_2(\rho, Obs) = \frac{1}{n+1} \cdot \sum_{i=0}^{n} |L(\delta(s_0, w_i))|$$

Maximal Cost of runs of length n:

$$\mathbf{Cost}_2(n, A, Obs) = \max\{\mathbf{Cost}_2(\rho, Obs) \textit{ for } \rho \in Runs^n(A)\}$$

The Cost of the pair (Obs, A) is

$$\mathbf{Cost}_2(A, Obs) = \limsup_{n \to \infty} \mathbf{Cost}_2(n, A, Obs)$$

Theorem

$\mathbf{Cost}_2(A, Obs)$ *can be computed in PTIME.*

How to compute the best or optimal observer ?

Computing the Cost of a Given Observer

$$\rho = q_0 \xrightarrow{a_1} q_1 \cdots q_{n-1} \xrightarrow{a_n} q_n \text{ a run of the plant } A$$

Let Obs be an observer and $w_i = \pi_{/\Sigma}(tr(q_0 \cdots q_i))$

$$\mathbf{Cost}_2(\rho, Obs) = \frac{1}{n+1} \cdot \sum_{i=0}^{n} |L(\delta(s_0, w_i)|$$

Maximal Cost of runs of length n:

$$\mathbf{Cost}_2(n, A, Obs) = \max\{\mathbf{Cost}_2(\rho, Obs) \textit{ for } \rho \in Runs^n(A)\}$$

The Cost of the pair (Obs, A) is

$$\mathbf{Cost}_2(A, Obs) = \limsup_{n \longrightarrow \infty} \mathbf{Cost}_2(n, A, Obs)$$

Theorem

$\mathbf{Cost}_2(A, Obs)$ *can be computed in PTIME.*

How to compute the best or optimal observer ?

Computing the Cost of a Given Observer

$$\rho = q_0 \xrightarrow{a_1} q_1 \cdots q_{n-1} \xrightarrow{a_n} q_n \text{ a run of the plant } A$$

Let Obs be an observer and $w_i = \pi_{/\Sigma}(tr(q_0 \cdots q_i))$

$$\mathbf{Cost}_2(\rho, Obs) = \frac{1}{n+1} \cdot \sum_{i=0}^{n} |L(\delta(s_0, w_i))|$$

Maximal Cost of runs of length n:

$$\mathbf{Cost}_2(n, A, Obs) = \max\{\mathbf{Cost}_2(\rho, Obs) \textit{ for } \rho \in Runs^n(A)\}$$

The Cost of the pair (Obs, A) is

$$\mathbf{Cost}_2(A, Obs) = \limsup_{n \longrightarrow \infty} \mathbf{Cost}_2(n, A, Obs)$$

Theorem

$\mathbf{Cost}_2(A, Obs)$ *can be computed in PTIME.*

How to compute the best or optimal observer ?

Computing the Cost of a Given Observer

$$\rho = q_0 \xrightarrow{a_1} q_1 \cdots q_{n-1} \xrightarrow{a_n} q_n \text{ a run of the plant } A$$

Let Obs be an observer and $w_i = \pi_{/\Sigma}(tr(q_0 \cdots q_i))$

$$\mathbf{Cost}_2(\rho, Obs) = \frac{1}{n+1} \cdot \sum_{i=0}^{n} |L(\delta(s_0, w_i))|$$

Maximal Cost of runs of length n:

$$\mathbf{Cost}_2(n, A, Obs) = \max\{\mathbf{Cost}_2(\rho, Obs) \textit{ for } \rho \in Runs^n(A)\}$$

The Cost of the pair (Obs, A) is

$$\mathbf{Cost}_2(A, Obs) = \limsup_{n \to \infty} \mathbf{Cost}_2(n, A, Obs)$$

Theorem

$\mathbf{Cost}_2(A, Obs)$ *can be computed in PTIME.*

How to compute the best or optimal observer ?

Computing the Cost of a Given Observer

$$\rho = q_0 \xrightarrow{a_1} q_1 \cdots q_{n-1} \xrightarrow{a_n} q_n \text{ a run of the plant } A$$

Let Obs be an observer and $w_i = \pi_{/\Sigma}(tr(q_0 \cdots q_i))$

$$\textbf{Cost}_2(\rho, Obs) = \frac{1}{n+1} \cdot \sum_{i=0}^{n} |L(\delta(s_0, w_i))|$$

Maximal Cost of runs of length n:

$$\textbf{Cost}_2(n, A, Obs) = \max\{\textbf{Cost}_2(\rho, Obs) \textit{ for } \rho \in Runs^n(A)\}$$

The Cost of the pair (Obs, A) is

$$\textbf{Cost}_2(A, Obs) = \limsup_{n \longrightarrow \infty} \textbf{Cost}_2(n, A, Obs)$$

Theorem

$\textbf{Cost}_2(A, Obs)$ *can be computed in PTIME.*

How to compute the best or optimal observer ?

Computing the Cost of a Given Observer

$$\rho = q_0 \xrightarrow{a_1} q_1 \cdots q_{n-1} \xrightarrow{a_n} q_n \text{ a run of the plant } A$$

Let Obs be an observer and $w_i = \pi_{/\Sigma}(tr(q_0 \cdots q_i))$

$$\mathbf{Cost}_2(\rho, Obs) = \frac{1}{n+1} \cdot \sum_{i=0}^{n} |L(\delta(s_0, w_i)|$$

Maximal Cost of runs of length n:

$$\mathbf{Cost}_2(n, A, Obs) = \max\{\mathbf{Cost}_2(\rho, Obs) \textit{ for } \rho \in Runs^n(A)\}$$

The Cost of the pair (Obs, A) is

$$\mathbf{Cost}_2(A, Obs) = \limsup_{n \to \infty} \mathbf{Cost}_2(n, A, Obs)$$

Theorem

$\mathbf{Cost}_2(A, Obs)$ *can be computed in PTIME.*

Use Karp's Maximum Mean-Weight Cycle Algorithm

Computing the Cost of a Given Observer

$$\rho = q_0 \xrightarrow{a_1} q_1 \cdots q_{n-1} \xrightarrow{a_n} q_n \text{ a run of the plant } A$$

Let Obs be an observer and $w_i = \pi_{/\Sigma}(tr(q_0 \cdots q_i))$

$$\mathbf{Cost}_2(\rho, Obs) = \frac{1}{n+1} \cdot \sum_{i=0}^{n} |L(\delta(s_0, w_i)|$$

Maximal Cost of runs of length n:

$$\mathbf{Cost}_2(n, A, Obs) = \max\{\mathbf{Cost}_2(\rho, Obs) \text{ for } \rho \in Runs^n(A)\}$$

The Cost of the pair (Obs, A) is

$$\mathbf{Cost}_2(A, Obs) = \limsup_{n \to \infty} \mathbf{Cost}_2(n, A, Obs)$$

Theorem

$\mathbf{Cost}_2(A, Obs)$ *can be computed in PTIME.*

How to compute the best or optimal observer ?

Bounded Cost Observer

Problem 2: Bounded Cost Observer

Input: A, $k \in \mathbb{N}$ and $c \in \mathbb{N}$.

Problem:

(A). Is there an observer Obs s.t. A is (Obs,k)-diagnosable and $\textbf{Cost}_2(\text{Obs}) \leq c$?

(B). If the answer to (A) is "yes", compute a witness observer Obs with $\textbf{Cost}_2(\text{Obs}) \leq c$.

Steps to Solve Problem 2:

- Step 1: Compute the most liberal observer O;
  i.e. obtain a representation of the set of all observers
- Step 2: Compute an optimal cost observer.

Theorem ([C. Tripakis Altisen, ACSD 2007])

There is a finite state most permissive observer for A.

Bounded Cost Observer

Problem 2: Bounded Cost Observer

Input: A, $k \in \mathbb{N}$ and $c \in \mathbb{N}$.
Problem:

(A). Is there an observer Obs s.t. A is (Obs,k)-diagnosable and **Cost**$_2$(Obs) $\leq c$?

(B). If the answer to (A) is "yes", compute a witness observer Obs with **Cost**$_2$(Obs) $\leq c$.

Steps to Solve Problem 2:

- Step 1: Compute the most liberal observer O; *i.e.* obtain a representation of the set of all observers
- Step 2: Compute an optimal cost observer.

Theorem ([C. Tripakis Altisen, ACSD 2007])

There is a finite state most permissive observer for A.

Bounded Cost Observer

Problem 2: Bounded Cost Observer

Input: A, $k \in \mathbb{N}$ and $c \in \mathbb{N}$.

Problem:

(A). Is there an observer Obs s.t. A is (Obs,k)-diagnosable and **Cost**$_2$(Obs) $\leq c$?

(B). If the answer to (A) is "yes", compute a witness observer Obs with **Cost**$_2$(Obs) $\leq c$.

Steps to Solve Problem 2:

- Step 1: Compute the most liberal observer O; *i.e.* obtain a representation of the set of all observers
- Step 2: Compute an optimal cost observer.

Theorem ([C. Tripakis Altisen, ACSD 2007])

There is a finite state most permissive observer for A.

Bounded Cost Observer

Problem 2: Bounded Cost Observer

Input: A, $k \in \mathbb{N}$ and $c \in \mathbb{N}$.
Problem:

(A). Is there an observer Obs s.t. A is (Obs,k)-diagnosable and **Cost**$_2$(Obs) $\leq c$?

(B). If the answer to (A) is "yes", compute a witness observer Obs with **Cost**$_2$(Obs) $\leq c$.

Steps to Solve Problem 2:

- Step 1: Compute the most liberal observer O;
  i.e. obtain a representation of the set of all observers
- Step 2: Compute an optimal cost observer.

Theorem ([C. Tripakis Altisen, ACSD 2007])

There is a finite state most permissive observer for A.

Bounded Cost Observer

Problem 2: Bounded Cost Observer

Input: A, $k \in \mathbb{N}$ and $c \in \mathbb{N}$.
Problem:

(A). Is there an observer Obs s.t. A is (Obs,k)-diagnosable and **Cost**$_2$(Obs) $\leq c$?

(B). If the answer to (A) is "yes", compute a witness observer Obs with **Cost**$_2$(Obs) $\leq c$.

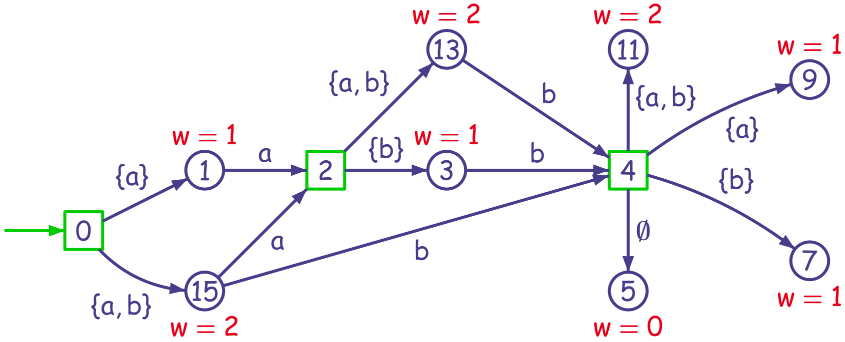
Steps to Solve Problem 2:

- Step 1: Compute the most liberal observer O;
  i.e. obtain a representation of the set of all observers
- Step 2: Compute an optimal cost observer.

Theorem ([C. Tripakis Altisen, ACSD 2007])

There is a finite state most permissive observer for A.

Optimal Dynamic Observer/Two-Player Game



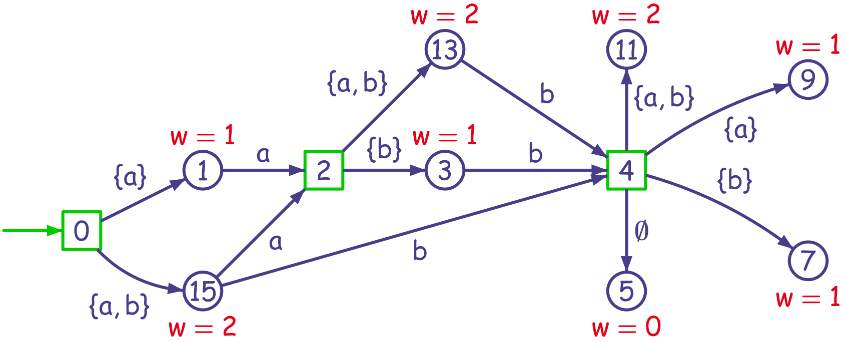
- Player 1 chooses what to observe: X / Player 2 generates $\rho.l$, $l \in X$
- Player 1 and 2 produce plays
- given a strategy for Player 1 and the moves of Player 2

$$w(\rho) \text{ is the sum of the weights } w_1 w_2 \cdots w_n \text{ of the play } \rho$$

$$Cost_2(\rho, Obs) = w(\rho)/(|\rho| + 1)$$

- Goal for Player 1: minimize $\limsup_{n \to \infty} \{ w(\rho)/(|\rho| + 1) \mid \rho \in Runs^n(A) \}$

Optimal Dynamic Observer/Two-Player Game



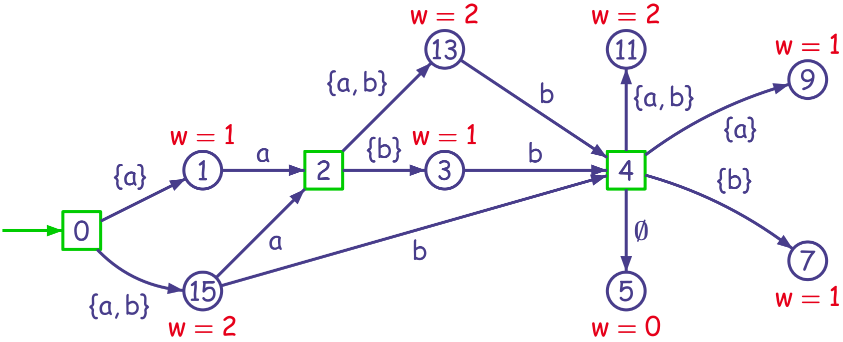
- Player 1 chooses what to observe: X / Player 2 generates $\rho.l$, $l \in X$
- Player 1 and 2 produce plays
- given a strategy for Player 1 and the moves of Player 2

$$w(\rho) \text{ is the sum of the weights } w_1 w_2 \cdots w_n \text{ of the play } \rho$$

$$Cost_2(\rho, Obs) = w(\rho)/(|\rho| + 1)$$

- Goal for Player 1: minimize $\limsup_{n \to \infty} \{w(\rho)/(|\rho| + 1) \mid \rho \in Runs^n(A)\}$

Optimal Dynamic Observer/Two-Player Game



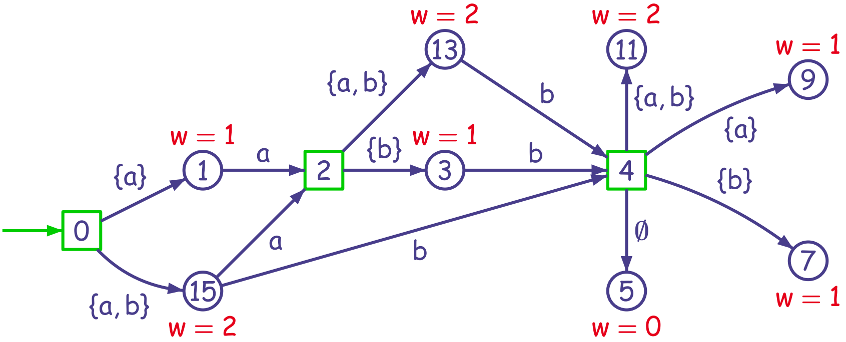
- Player 1 chooses what to observe: X / Player 2 generates $\rho.l$, $l \in X$
- Player 1 and 2 produce plays
- given a strategy for Player 1 and the moves of Player 2

  $w(\rho)$ is the sum of the weights $w_1 w_2 \cdots w_n$ of the play ρ

  $$Cost_2(\rho, Obs) = w(\rho)/(|\rho| + 1)$$

- Goal for Player 1: minimize $\limsup_{n\to\infty}\{w(\rho)/(|\rho|+1) \mid \rho \in Runs^n(A)\}$

Optimal Dynamic Observer/Two-Player Game



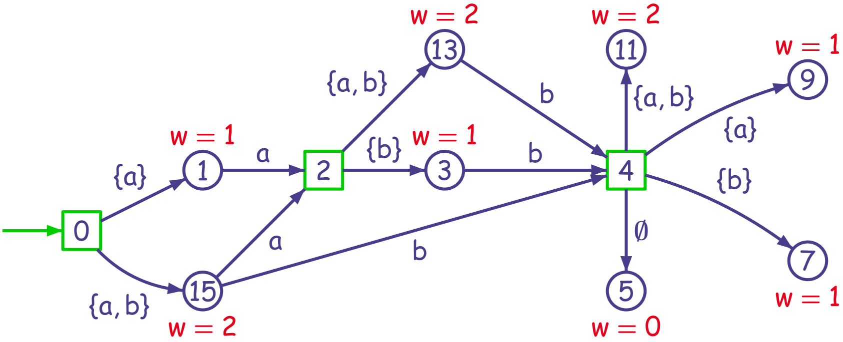
- Player 1 chooses what to observe: X / Player 2 generates $\rho.l$, $l \in X$
- Player 1 and 2 produce plays
- given a strategy for Player 1 and the moves of Player 2

$$w(\rho) \text{ is the sum of the weights } w_1 w_2 \cdots w_n \text{ of the play } \rho$$

$$Cost_2(\rho, Obs) = w(\rho)/(|\rho| + 1)$$

- Goal for Player 1: minimize $\limsup_{n \to \infty}\{w(\rho)/(|\rho|+1) \mid \rho \in Runs^n(A)\}$

Optimal Dynamic Observer/Two-Player Game



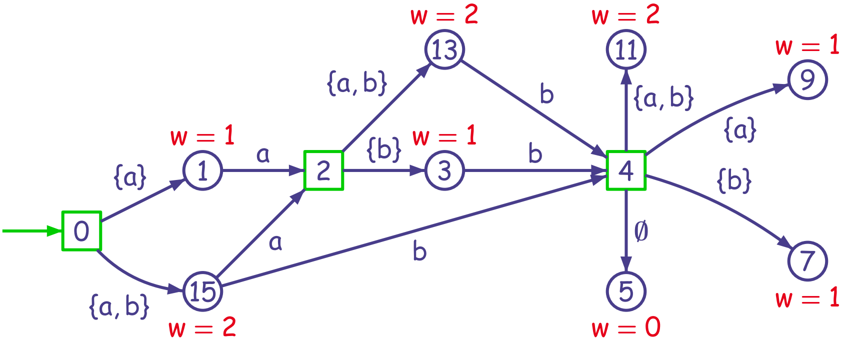
- Player 1 chooses what to observe: X / Player 2 generates $\rho.l$, $l \in X$
- Player 1 and 2 produce plays
- given a strategy for Player 1 and the moves of Player 2

  $w(\rho)$ is the sum of the weights $w_1 w_2 \cdots w_n$ of the play ρ

  $$\mathbf{Cost}_2(\rho, Obs) = w(\rho)/(|\rho| + 1)$$

- Goal for Player 1: minimize $\limsup_{n\to\infty}\{w(\rho)/(|\rho|+1) \mid \rho \in Runs^n(A)\}$

Optimal Dynamic Observer/Two-Player Game



- Player 1 chooses what to observe: X / Player 2 generates $\rho.l$, $l \in X$
- Player 1 and 2 produce plays
- given a strategy for Player 1 and the moves of Player 2

$$w(\rho) \text{ is the sum of the weights } w_1 w_2 \cdots w_n \text{ of the play } \rho$$

$$\mathbf{Cost}_2(\rho, Obs) = w(\rho)/(|\rho| + 1)$$

- Goal for Player 1: minimize $\limsup_{n \to \infty} \{w(\rho)/(|\rho| + 1) \,|\, \rho \in Runs^n(A)\}$

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight w(s)
  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer O
  2. Build a weighted graph game: $O \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight w(s)
  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer O
  2. Build a weighted graph game: $O \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight w(s)
  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer O
  2. Build a weighted graph game: $O \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight w(s)
  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer $\mathcal{O}$
  2. Build a weighted graph game: $\mathcal{O} \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight $w(s)$
  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer O
  2. Build a weighted graph game: $O \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight w(s)
  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer O
  2. Build a weighted graph game: $O \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight $w(s)$
  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer O
  2. Build a weighted graph game: $O \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight $w(s)$
  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer O
  2. Build a weighted graph game: $O \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight w(s)

  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer O
  2. Build a weighted graph game: $O \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer

Mean Payoff Games

[Zwick & Paterson, TCS 1996]

- Weighted two-player games
- Each state s has a weight w(s)
  can be done with weight on edges
- Goal of the Players:
  - Player 1: minimize $l_0 = \limsup w(\rho)/(|\rho| + 1)$
  - Player 2: maximize $l_1 = \liminf w(\rho)/(|\rho| + 1)$
- Results for weighted two-player games:
  - $\exists v \in \mathbb{Q}$ s.t. each player has a memoryless strategy to ensure $l_0 \leq v$ and $l_1 \geq v$
  - v can be effectively computed
  - Memoryless strategies for both players can be effectively computed
- Solution to Optimal Cost Dynamic Observer Problem:
  1. Compute the most liberal observer O
  2. Build a weighted graph game: $O \times A$
  3. Use Zwick & Paterson's algorithm

Winning Strategy for Player 1 = Optimal Observer