

The Complexity of Codiagnosability for Discrete Event and Timed Systems

Franck Cassez

**National ICT Australia & CNRS
Sydney, Australia**

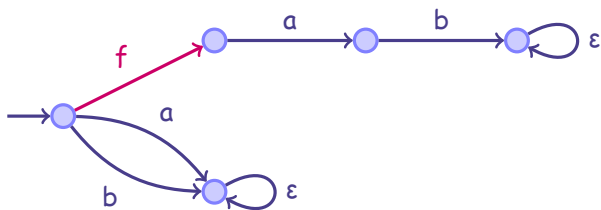
**September 24th, 2010
ATVA'2010, Singapore**



Outline of the talk

- 1 **Fault Diagnosis**
- 2 **Fault Codiagnosis**
- 3 **Complexity of Codiagnosis Problems**
- 4 **Synthesis of Codiagnosers for Timed Systems**
- 5 **Conclusion**

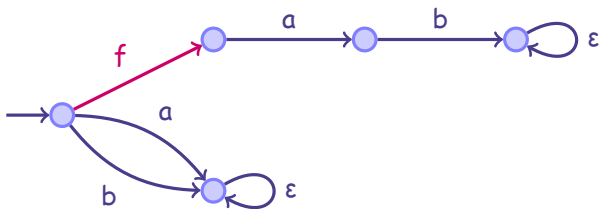
Fault Diagnosis for Discrete Event Systems



Given:

- A finite automaton A over $\Sigma^{\epsilon, f} = \Sigma \cup \{\epsilon, f\}$
- f is the fault action, Σ is the set of observable events

Fault Diagnosis for Discrete Event Systems



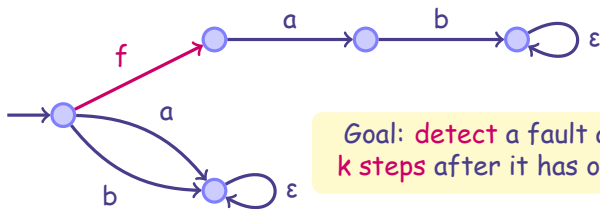
Given:

- A finite automaton A over $\Sigma^{\varepsilon, f} = \Sigma \cup \{\varepsilon, f\}$
- f is the fault action, Σ is the set of observable events

Notations:

- **Faulty** _{$\geq k$} (A): k -faulty runs that contain f followed by $\geq k$ actions
- **NonFaulty**(A): Non faulty runs that contain no f

Fault Diagnosis for Discrete Event Systems



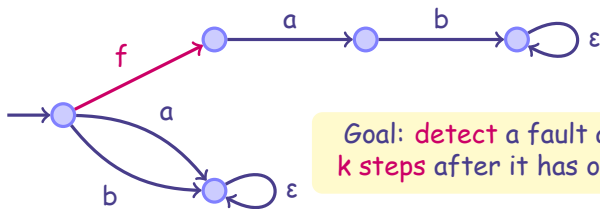
Given:

- A finite automaton A over $\Sigma^{\epsilon, f} = \Sigma \cup \{\epsilon, f\}$
- f is the fault action, Σ is the set of observable events

Notations:

- **Faulty** _{$\geq k$} (A): k -faulty runs that contain f followed by $\geq k$ actions
- **NonFaulty**(A): Non faulty runs that contain no f

Fault Diagnosis for Discrete Event Systems



Given:

- A finite automaton A over $\Sigma^{\epsilon, f} = \Sigma \cup \{\epsilon, f\}$
- f is the fault action, Σ is the set of observable events

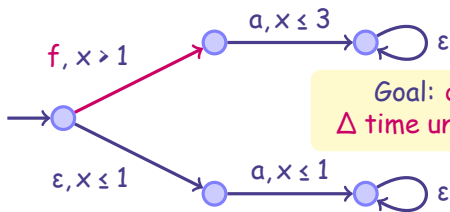
Notations:

- **Faulty** _{$\geq k$} (A): k -faulty runs that contain f followed by $\geq k$ actions
- **NonFaulty**(A): Non faulty runs that contain no f

Fault diagnosis in discrete-time: given k , and observable events Σ

- **never** raise an alarm on non-faulty runs
- **always** raise an alarm on k -faulty runs

Fault Diagnosis for Dense-Time Systems



Goal: detect a fault at most Δ time units after it has occurred

Given:

- A timed automaton with continuous variables A over $\Sigma^{\epsilon, f} = \Sigma \cup \{f\}$
- f is the fault action, Σ is the set of observable events

Notations:

- **Faulty** $_{\geq \Delta}(A)$: Δ -faulty runs that contain f followed by $\geq \Delta$ time units
- **NonFaulty** (A) : Non faulty runs (contain no f)

Fault diagnosis in dense-time: given Δ , and observable events Σ

- never raise an alarm on non-faulty runs
- always raise an alarm on Δ -faulty runs

Diagnosability Problems

$\text{trace}(\rho)$ = trace of the run ρ (a word in $(\Sigma \cup \{\varepsilon, f\})^*$)

$\pi_\Sigma(\text{trace}(\rho))$ = projection of the trace on **observable events**

Definition (k-diagnoser)

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

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

k-Diagnosability Problem

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

Diagnosability Problem

Given A , is there a $k \in \mathbb{N}$ s.t. A is **k-diagnosable**?

Dense-time version defined using **timed words**, and **timed languages**

Diagnosability Problems

$\text{trace}(\rho)$ = trace of the run ρ (a word in $(\Sigma \cup \{\varepsilon, f\})^*$)

$\pi_\Sigma(\text{trace}(\rho))$ = projection of the trace on **observable events**

Definition (k-diagnoser)

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

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

k-Diagnosability Problem

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

Diagnosability Problem

Given A , is there a $k \in \mathbb{N}$ s.t. A is **k-diagnosable**?

Dense-time version defined using **timed words**, and **timed languages**

Diagnosability Problems

$\text{trace}(\rho)$ = trace of the run ρ (a word in $(\Sigma \cup \{\varepsilon, f\})^*$)

$\pi_{\Sigma}(\text{trace}(\rho))$ = projection of the trace on **observable events**

Definition (k-diagnoser)

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

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

k-Diagnosability Problem

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

Diagnosability Problem

Given A , is there a $k \in \mathbb{N}$ s.t. A is **k-diagnosable**?

Dense-time version defined using **timed words**, and **timed languages**

Diagnosability Problems

$\text{trace}(\rho)$ = trace of the run ρ (a word in $(\Sigma \cup \{\varepsilon, f\})^*$)

$\pi_\Sigma(\text{trace}(\rho))$ = projection of the trace on **observable events**

Definition (k-diagnoser)

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

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

k-Diagnosability Problem

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

Diagnosability Problem

Given A , is there a $k \in \mathbb{N}$ s.t. A is **k-diagnosable**?

Dense-time version defined using **timed words**, and **timed languages**

Diagnosability Problems

$\text{trace}(\rho)$ = trace of the run ρ (a word in $(\Sigma \cup \{\varepsilon, f\})^*$)

$\pi_\Sigma(\text{trace}(\rho))$ = projection of the trace on **observable events**

Definition (k-diagnoser)

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

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

k-Diagnosability Problem

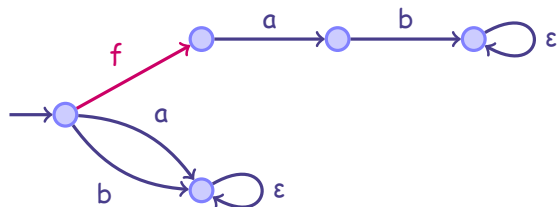
Given A and $k \in \mathbb{N}$, is there a **k-diagnoser** for A ?

Diagnosability Problem

Given A , is there a $k \in \mathbb{N}$ s.t. A is **k-diagnosable**?

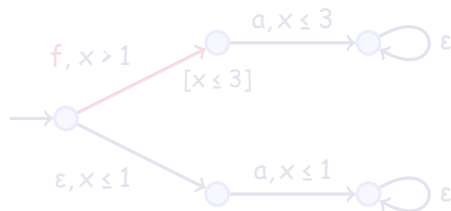
Dense-time version defined using **timed words**, and **timed languages**

Examples



1-diagnosable ? No

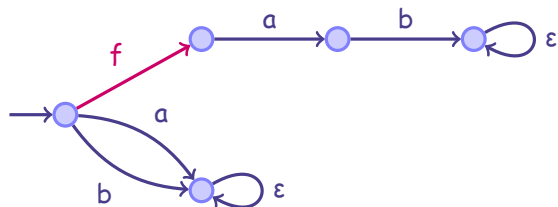
2-diagnosable ? Yes



2-diagnosable ? Yes

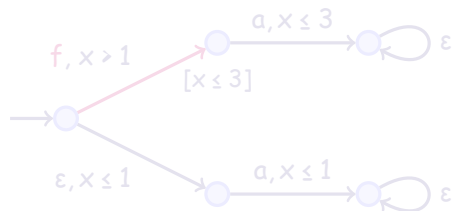
1-diagnosable ? No

Examples



1-diagnosable ? **No**

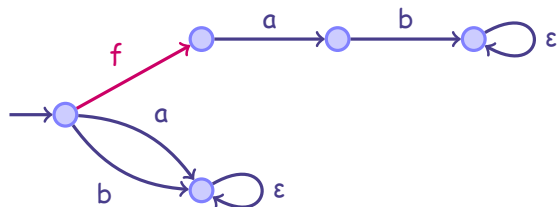
2-diagnosable ? *Yes*



2-diagnosable ? *Yes*

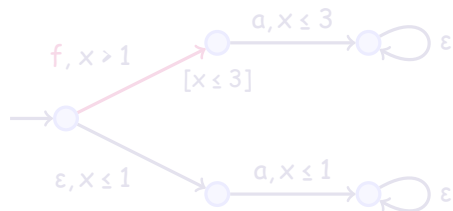
1-diagnosable ? *No*

Examples



1-diagnosable ? No

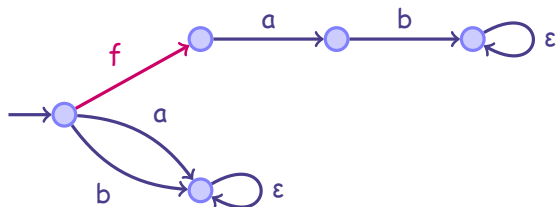
2-diagnosable ? **Yes**



2-diagnosable ? **Yes**

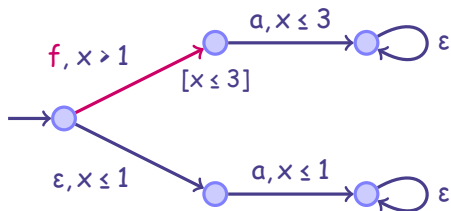
1-diagnosable ? No

Examples



1-diagnosable ? No

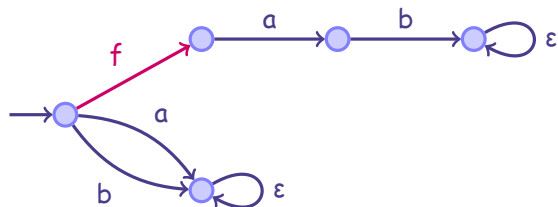
2-diagnosable ? Yes



2-diagnosable ? Yes

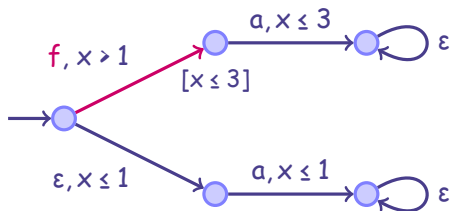
1-diagnosable ? No

Examples



1-diagnosable ? No

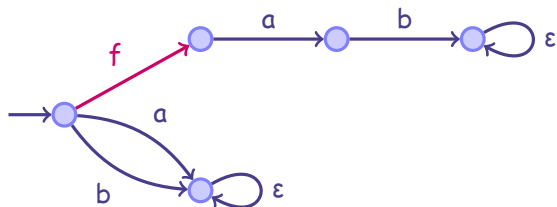
2-diagnosable ? Yes



2-diagnosable ? **Yes**

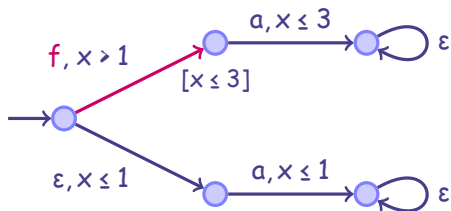
1-diagnosable ? No

Examples



1-diagnosable ? No

2-diagnosable ? Yes



2-diagnosable ? Yes

1-diagnosable ? No

Results for the Diagnosis Problems

Diagnosis Problems

- k-diagnosability (DES) or Δ -diagnosability (TA)
- Diagnosability: $\exists k$ s.t. a system is k-diagnosable ?
- Optimal Delay: compute the **minimum k**
- Synthesis Problem: **compute a diagnoser**

Results

Discrete Event Systems

k-diagnosability	diagnosability	Optimal Delay	Synthesis
P TIME	P TIME	P TIME	EX PTIME
[Yoo 2002] [Jiang 2001]	[Yoo 2002] [Jiang 2001]	[Yoo 2002] [Jiang 2001]	[Sampath 1995]

Timed Systems (Timed Automata)

Δ -diagnosability	diagnosability	Optimal Delay	Synthesis
P SPACE- C.	P SPACE- C.	P SPACE	2EX PTIME- C.
[Tripakis 2002]	[Tripakis 2002]	[C. 2009]	[Bouyer 2005]

Results for the Diagnosis Problems

Diagnosis Problems

- k-diagnosability (DES) or Δ -diagnosability (TA)
- Diagnosability: $\exists k$ s.t. a system is k-diagnosable ?
- Optimal Delay: compute the **minimum k**
- Synthesis Problem: **compute a diagnoser**

Results

Discrete Event Systems

k-diagnosability	diagnosability	Optimal Delay	Synthesis
P TIME	P TIME	P TIME	EX PTIME
[Yoo 2002] [Jiang 2001]	[Yoo 2002] [Jiang 2001]	[Yoo 2002] [Jiang 2001]	[Sampath 1995]

Timed Systems (Timed Automata)

Δ -diagnosability	diagnosability	Optimal Delay	Synthesis
PSPACE-C.	PSPACE-C.	PSPACE	2EXPTIME-C.
[Tripakis 2002]	[Tripakis 2002]	[C. 2009]	[Bouyer 2005]

Fault Codiagnosis

Decentralized Fault Diagnosis

- **Generalization** of diagnosability for **decentralized architectures**
- **local observations + coordination**
(a coordinator can compute a result from distributed information)
- Various notions of decentralized architectures [Debouk 2000]
- **Codiagnosability**: no communication between local observers
⇒ each fault can be detected by at least one local observer

(Lack of) Results

Discrete Event Systems

k-codiagnosability	Codiagnosability	Optimal Delay	Synthesis
??	EXPTIME [Wang 2007]	??	??

Timed Systems (Timed Automata)

Δ -codiagnosability	Codiagnosability	Optimal Delay	Synthesis
??	??	??	??

Fault Codiagnosis

Decentralized Fault Diagnosis

- **Generalization** of diagnosability for **decentralized architectures**
- **local observations + coordination**
(a coordinator can compute a result from distributed information)
- Various notions of decentralized architectures [Debouk 2000]
- **Codiagnosability**: no communication between local observers
⇒ each fault can be detected by at least one local observer

(Lack of) Results

Discrete Event Systems

k-codiagnosability	Codiagnosability	Optimal Delay	Synthesis
??	EXPTIME [Wang 2007]	??	??

Timed Systems (Timed Automata)

Δ -codiagnosability	Codiagnosability	Optimal Delay	Synthesis
??	??	??	??

Our Contribution

Characterization of Codiagnosability

- generalize DES condition for diagnosability to codiagnosability
- extend to timed systems (timed automata)

Codiagnosability Problems

- 1 settle exact complexity of codiagnosis problems for DES
e.g., PSPACE-C. vs EXPTIME for Codiagnosability
- 2 address codiagnosis problems for TA and settle complexity

Synthesis Problem

- For DES: easy extension of earlier results for DES [Sampath 1995]
- for TA: two problems depending on the **resources** of the diagnoser
 - ▶ unbounded: easy extension of "diagnosis" result [Tripakis 2002]
 - ▶ bounded: extension of result in [Bouyer 2005]

Codiagnosers & Codiagnosis Problems

A : timed automaton over the alphabet $\Sigma_{\varepsilon, f}$

$\Delta \in \mathbb{N}$

$\mathcal{E} = (\Sigma_i)_{1 \leq i \leq n}$ a family of subsets of Σ

$TW^*(\Sigma_i)$ timed words over Σ_i e.g., a 1.32 b π a 4.57 ...

Definition ((Δ, \mathcal{E})-Codiagnoser)

A (Δ, \mathcal{E})-codiagnoser for A is a mapping $\vec{D} = (D_1, \dots, D_n)$ with $D_i : TW^*(\Sigma_i) \rightarrow \{0, 1\}$ such that:

- for each $\rho \in \text{NonFaulty}(A)$, $\sum_{i=1}^n \vec{D}[i](\pi_i(\text{trace}(\rho))) = 0$,
- for each $\rho \in \text{Faulty}_{\geq \Delta}(A)$, $\sum_{i=1}^n \vec{D}[i](\pi_i(\text{trace}(\rho))) \geq 1$.

Codiagnosis Problems

- 1 (Δ, \mathcal{E})-codiagnosability: is there any (Δ, \mathcal{E})-codiagnoser for A ?
- 2 \mathcal{E} -codiagnosability: is there any $\Delta \in \mathbb{N}$ s.t. A is (Δ, \mathcal{E})-codiagnosable?
- 3 Optimal delay: what is the minimum Δ ?
- 4 Synthesis: can we compute a codiagnoser?

Necess. & Sufficient Condition for Codiagnosability

π_Σ projection on Σ

π_i projection on Σ_i

$\pi_i^{-1}(w) = \{w' \in TW^*(\Sigma) \mid \pi_i(w') = w\}$ (inverse projection)

Necessary and Sufficient Condition for Diagnosability

A is **not** (Δ, Σ) -diagnosable if and only if $\exists \varrho \in \mathbf{Faulty}_{\Sigma, \Delta}(A)$ and

$$\forall 1 \leq i \leq n, \exists \varrho' \in \mathbf{NonFaulty}(A) \text{ s.t. } \pi_\Sigma(\text{trace}(\varrho)) = \pi_\Sigma(\text{trace}(\varrho'))$$

Language-based characterization of diagnosability

Necess. & Sufficient Condition for Codiagnosability

π_Σ projection on Σ

π_i projection on Σ_i

$\pi_i^{-1}(w) = \{w' \in TW^*(\Sigma) \mid \pi_i(w') = w\}$ (inverse projection)

Necessary and Sufficient Condition for Diagnosability

A is **not** (Δ, Σ) -diagnosable if and only if $\exists \varrho \in \mathbf{Faulty}_{\geq \Delta}(A)$ and

$$\forall 1 \leq i \leq n, \exists \varrho' \in \mathbf{NonFaulty}(A) \text{ s.t. } \pi_\Sigma(\text{trace}(\varrho)) = \pi_\Sigma(\text{trace}(\varrho'))$$

Language-based characterization of diagnosability

$$\mathbf{Faulty}_{\geq \Delta}^{\text{trace}}(A) \cap \mathbf{NonFaulty}^{\text{trace}}(A) = \emptyset$$

Necess. & Sufficient Condition for Codiagnosability

π_Σ projection on Σ

π_i projection on Σ_i

$\pi_i^{-1}(w) = \{w' \in TW^*(\Sigma) \mid \pi_i(w') = w\}$ (inverse projection)

Necessary and Sufficient Condition for Codiagnosability

A is not (Δ, \mathcal{E}) -codiagnosable if and only if $\exists \varrho \in \mathbf{Faulty}_{\geq \Delta}(A)$ and

$$\forall 1 \leq i \leq n, \exists \varrho_i \in \mathbf{NonFaulty}(A) \text{ s.t. } \pi_i(\text{trace}(\varrho)) = \pi_i(\text{trace}(\varrho_i))$$

Language-based characterization of codiagnosability

Necess. & Sufficient Condition for Codiagnosability

π_Σ projection on Σ

π_i projection on Σ_i

$\pi_i^{-1}(w) = \{w' \in TW^*(\Sigma) \mid \pi_i(w') = w\}$ (inverse projection)

Necessary and Sufficient Condition for Codiagnosability

A is not (Δ, \mathcal{E}) -codiagnosable if and only if $\exists \varrho \in \mathbf{Faulty}_{\geq \Delta}(A)$ and

$$\forall 1 \leq i \leq n, \exists \varrho_i \in \mathbf{NonFaulty}(A) \text{ s.t. } \pi_i(\text{trace}(\varrho)) = \pi_i(\text{trace}(\varrho_i))$$

Language-based characterization of codiagnosability

$$\mathbf{Faulty}_{\geq \Delta}^{\text{trace}}(A) \cap \left(\bigcap_{i=1}^n \pi_i^{-1}(\pi_i(\mathbf{NonFaulty}^{\text{trace}}(A))) \right) = \emptyset \quad (\text{NSC})$$

Complexity of Codiagnosability

$\mathcal{E} = (\Sigma_i), 1 \leq i \leq n$ a family of subsets of Σ

(Δ, \mathcal{E}) -Codiagnosability

Input: a timed automaton A , $\Delta \in \mathbb{N}$

Problem: Is A (Δ, \mathcal{E}) -Codiagnosable?

Size of input: $|A| + \log \Delta + \sum_{i=1}^n |\Sigma_i| \leq |A| + \log \Delta + n \cdot |\Sigma|$

Solution: Check condition (NSC)

$$\text{Faulty}_{\geq \Delta}^{\text{trace}}(A) \cap \left(\bigcap_{i=1}^n \pi_i^{-1}(\pi_i(\text{NonFaulty}^{\text{trace}}(A))) \right) = \emptyset$$

(Δ, \mathcal{E}) -Codiagnosability *reduces* to intersection emptiness for TA

Complexity of Codiagnosability

$\mathcal{E} = (\Sigma_i), 1 \leq i \leq n$ a family of subsets of Σ

(Δ, \mathcal{E}) -Codiagnosability

Input: a timed automaton A , $\Delta \in \mathbb{N}$

Problem: Is A (Δ, \mathcal{E}) -Codiagnosable ?

Size of input: $|A| + \log \Delta + \sum_{i=1}^n |\Sigma_i| \leq |A| + \log \Delta + n \cdot |\Sigma|$

Solution: Check condition (NSC)

$$\text{Faulty}_{\geq \Delta}^{\text{trace}}(A) \cap \left(\bigcap_{i=1}^n \pi_i^{-1}(\pi_i(\text{NonFaulty}^{\text{trace}}(A))) \right) = \emptyset$$

Algorithm

- 1 Build A_i^* s.t. $\mathcal{L}(A_i^*) = \pi_i^{-1}(\pi_i(\text{NonFaulty}^{\text{trace}}(A)))$
- 2 Check **emptiness** of $A_{\geq \Delta}^{\text{trace}} \cap A_1^* \cap A_2^* \cap \dots \cap A_n^*$

Complexity of Codiagnosability

$\mathcal{E} = (\Sigma_i), 1 \leq i \leq n$ a family of subsets of Σ

(Δ, \mathcal{E}) -Codiagnosability

Input: a timed automaton A , $\Delta \in \mathbb{N}$

Problem: Is A (Δ, \mathcal{E}) -Codiagnosable?

Size of input: $|A| + \log \Delta + \sum_{i=1}^n |\Sigma_i| \leq |A| + \log \Delta + n \cdot |\Sigma|$

Solution: Check condition (NSC)

$$\text{Faulty}_{\geq \Delta}^{\text{trace}}(A) \cap \left(\bigcap_{i=1}^n \pi_i^{-1}(\pi_i(\text{NonFaulty}^{\text{trace}}(A))) \right) = \emptyset$$

(Δ, \mathcal{E}) -Codiagnosability *reduces* to intersection emptiness for TA

Complexity of Codiagnosability (cont'd)

Theorem (Intersection Emptiness for TA)

For $(A_i)_{1 \leq i \leq n}$, checking whether $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$ is PSPACE-complete.

Hardness: $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$? PSPACE-hard for Deterministic Finite Automata where only A_1 has a set of accepting states

For DFA, Intersection Emptiness reduces to codiagnosability [▶ Proof](#)

Theorem $((\Delta, \mathcal{E})$ -Codiagnosability)

(Δ, \mathcal{E}) -Codiagnosability is PSPACE-complete for Timed Automata and already PSPACE-hard for Deterministic Finite Automata.

Theorem $(\mathcal{E}$ -Codiagnosability)

\mathcal{E} -Codiagnosability is PSPACE-complete for Timed Automata and already PSPACE-hard for Deterministic Finite Automata.

Complexity of Codiagnosability (cont'd)

Theorem (Intersection Emptiness for TA)

For $(A_i)_{1 \leq i \leq n}$, checking whether $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$ is PSPACE-complete.

Hardness: $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$? PSPACE-hard for Deterministic Finite Automata where only A_1 has a set of accepting states

For DFA, Intersection Emptiness reduces to codiagnosability [▶ Proof](#)

Theorem $((\Delta, \mathcal{E})$ -Codiagnosability)

(Δ, \mathcal{E}) -Codiagnosability is PSPACE-complete for Timed Automata and already PSPACE-hard for Deterministic Finite Automata.

Theorem $(\mathcal{E}$ -Codiagnosability)

\mathcal{E} -Codiagnosability is PSPACE-complete for Timed Automata and already PSPACE-hard for Deterministic Finite Automata.

Complexity of Codiagnosability (cont'd)

Theorem (Intersection Emptiness for TA)

For $(A_i)_{1 \leq i \leq n}$, checking whether $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$ is PSPACE-complete.

Hardness: $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$? PSPACE-hard for Deterministic Finite Automata where only A_1 has a set of accepting states

For DFA, Intersection Emptiness reduces to codiagnosability [▶ Proof](#)

Theorem $((\Delta, \varepsilon)$ -Codiagnosability)

(Δ, ε) -Codiagnosability is PSPACE-complete for Timed Automata and already PSPACE-hard for Deterministic Finite Automata.

Theorem (ε) -Codiagnosability)

ε -Codiagnosability is PSPACE-complete for Timed Automata and already PSPACE-hard for Deterministic Finite Automata.

Complexity of Codiagnosability (cont'd)

Theorem (Intersection Emptiness for TA)

For $(A_i)_{1 \leq i \leq n}$, checking whether $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$ is PSPACE-complete.

Hardness: $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$? PSPACE-hard for Deterministic Finite Automata where only A_1 has a set of accepting states

For DFA, Intersection Emptiness reduces to codiagnosability [▶ Proof](#)

Theorem ((Δ, ε) -Codiagnosability)

(Δ, ε) -Codiagnosability is PSPACE-complete for Timed Automata and already PSPACE-hard for Deterministic Finite Automata.

Theorem (ε -Codiagnosability)

ε -Codiagnosability is PSPACE-complete for Timed Automata and already PSPACE-hard for Deterministic Finite Automata.

Optimal Delay and Synthesis

Optimal Delay

Theorem (Optimal Delay)

Optimal Delay can be computed in PSPACE for FA and TA.

Synthesis of Codiagnosers

- Discrete Event Systems: generalization of [Sampath 1995]
 - determinize automata $B_i, 1 \leq i \leq n$ accepting $\pi_i(\text{Faulty}_{\text{ex}}(A))$
 - codiagnoser of exponential size
- Timed Automata: generalization of [Tripakis 2002]
 - for each $1 \leq i \leq n$, compute on-the-fly the states estimate E_i of A after reading $\pi_i(w)$ for $w \in \text{TW}^*(A)$
 - if each state in E_i is faulty output 1, otherwise output 0
 - Codiagnoser is a Turing Machine

[Bouyer 2005]: Bounded Resources Synthesis Problem for diagnosis of TA

Is there a codiagnoser which is a tuple of DTA?

Optimal Delay and Synthesis

Optimal Delay

Theorem (Optimal Delay)

Optimal Delay can be computed in PSPACE for FA and TA.

Synthesis of Codiagnosers

- Discrete Event Systems: generalization of [Sampath 1995]
 - determinize automata $B_i, 1 \leq i \leq n$ accepting $\pi_i(\text{Faulty}_{\text{ex}}(A))$
 - codiagnoser of exponential size
- Timed Automata: generalization of [Tripakis 2002]
 - for each $1 \leq i \leq n$, compute on-the-fly the states estimate E_i of A after reading $\pi_i(w)$ for $w \in \text{TW}^*(A)$
 - if each state in E_i is faulty output 1, otherwise output 0
 - Codiagnoser is a Turing Machine

[Bouyer 2005]: Bounded Resources Synthesis Problem for diagnosis of TA

Is there a codiagnoser which is a tuple of DTA?

Optimal Delay and Synthesis

Optimal Delay

Theorem (Optimal Delay)

Optimal Delay can be computed in PSPACE for FA and TA.

Synthesis of Codiagnosers

- Discrete Event Systems: generalization of [Sampath 1995]
 - ▶ **determinize** automata $B_i, 1 \leq i \leq n$ accepting $\pi_i(\text{Faulty}_{z_k}(A))$
 - ▶ codiagnoser of **exponential size**
- Timed Automata: generalization of [Tripakis 2002]
 - ▶ for each $1 \leq i \leq n$, compute on-the-fly the states estimate E_i of A after reading $\pi_i(w)$ for $w \in TW^*(A)$
 - ▶ if each state in E_i is faulty output 1, otherwise output 0
 - ▶ Codiagnoser is a Turing Machine

[Bouyer 2005]: **Bounded Resources Synthesis Problem** for diagnosis of TA

Is there a **codiagnoser** which is a **tuple of DTA**?

Optimal Delay and Synthesis

Optimal Delay

Theorem (Optimal Delay)

Optimal Delay can be computed in PSPACE for FA and TA.

Synthesis of Codiagnosers

- Discrete Event Systems: generalization of [Sampath 1995]
 - ▶ **determinize** automata $B_i, 1 \leq i \leq n$ accepting $\pi_i(\text{Faulty}_{\geq k}(A))$
 - ▶ codiagnoser of **exponential size**
- Timed Automata: generalization of [Tripakis 2002]
 - ▶ for each $1 \leq i \leq n$, compute **on-the-fly** the states estimate E_i of A after reading $\pi_i(w)$ for $w \in TW^*(A)$
 - ▶ if each state in E_i is **faulty** output 1, otherwise output 0
 - ▶ Codiagnoser is a **Turing Machine**

[Bouyer 2005]: **Bounded Resources Synthesis Problem** for diagnosis of TA

Is there a **codiagnoser** which is a **tuple of DTA**?

Optimal Delay and Synthesis

Optimal Delay

Theorem (Optimal Delay)

Optimal Delay can be computed in PSPACE for FA and TA.

Synthesis of Codiagnosers

- Discrete Event Systems: generalization of [Sampath 1995]
 - ▶ **determinize** automata $B_i, 1 \leq i \leq n$ accepting $\pi_i(\text{Faulty}_{\geq k}(A))$
 - ▶ codiagnoser of **exponential size**
- Timed Automata: generalization of [Tripakis 2002]
 - ▶ for each $1 \leq i \leq n$, compute **on-the-fly** the states estimate E_i of A after reading $\pi_i(w)$ for $w \in TW^*(A)$
 - ▶ if each state in E_i is **faulty** output 1, otherwise output 0
 - ▶ Codiagnoser is a **Turing Machine**

[Bouyer 2005]: **Bounded Resources Synthesis** Problem for diagnosis of TA

Is there a **codiagnoser** which is a **tuple of DTA**?

Optimal Delay and Synthesis

Optimal Delay

Theorem (Optimal Delay)

Optimal Delay can be computed in PSPACE for FA and TA.

Synthesis of Codiagnosers

- Discrete Event Systems: generalization of [Sampath 1995]
 - ▶ **determinize** automata B_i , $1 \leq i \leq n$ accepting $\pi_i(\text{Faulty}_{z_k}(A))$
 - ▶ codiagnoser of **exponential size**
- Timed Automata: generalization of [Tripakis 2002]
 - ▶ for each $1 \leq i \leq n$, compute **on-the-fly** the states estimate E_i of A after reading $\pi_i(w)$ for $w \in TW^*(A)$
 - ▶ if each state in E_i is **faulty** output 1, otherwise output 0
 - ▶ Codiagnoser is a **Turing Machine**

[Bouyer 2005]: **Bounded Resources Synthesis** Problem for diagnosis of TA

Is there a **codiagnoser** which is a **tuple of DTA**?

Fault Diagnosis with Deterministic TA [Bouyer 2005]

Δ -DTA-Diagnosability = Δ -diagnosability + **diagnoser** is a DTA

Resources: number of clocks + maximal constant + granularity + alphabet
resource $\mu = (\{x, y\}, 2, \frac{1}{3}, \{a\})$

Δ -DTA- μ -diagnosability

Input: A timed automaton A , $\Delta \in \mathbb{N}$, resource μ .

Problem: Is there any DTA D of resource μ s.t. A is (Δ, D) -diagnosable ?

Theorem [Bouyer 2005]

Δ -DTA- μ -diagnosability is 2EXPTIME-complete.

Δ -DTA- $\vec{\mu}$ -Codiagnosability

Input: a TA A , $\Delta \in \mathbb{N}$, and a **family of resources** $\vec{\mu} = (\mu_i)_{1 \leq i \leq n}$ with $\Sigma_i \subseteq \Sigma$.

Problem: Is there any codiagnoser $\vec{D} = (D_1, D_2, \dots, D_n)$ with $D_i \in \text{DTA}_{\mu_i}$ s.t. A is (Δ, \vec{D}) -codiagnosable ?

Fault Diagnosis with Deterministic TA [Bouyer 2005]

Δ -DTA-Diagnosability = Δ -diagnosability + **diagnoser** is a DTA

Resources: number of clocks + maximal constant + granularity + alphabet
resource $\mu = (\{x, y\}, 2, \frac{1}{3}, \{a\})$

Δ -DTA- μ -diagnosability

Input: A timed automaton A , $\Delta \in \mathbb{N}$, resource μ .

Problem: Is there any DTA D of resource μ s.t. A is (Δ, D) -diagnosable ?

Theorem [Bouyer 2005]

Δ -DTA- μ -diagnosability is 2EXPTIME-complete.

Δ -DTA- $\vec{\mu}$ -Codiagnosability

Input: a TA A , $\Delta \in \mathbb{N}$, and a **family of resources** $\vec{\mu} = (\mu_i)_{1 \leq i \leq n}$ with $\Sigma_i \subseteq \Sigma$.

Problem: Is there any codiagnoser $\vec{D} = (D_1, D_2, \dots, D_n)$ with $D_i \in \text{DTA}_{\mu_i}$ s.t. A is (Δ, \vec{D}) -codiagnosable ?

Synthesis of a DTA Codiagnoser

Algorithm [Bouyer 2005]

- 1 Construct a two-player game finite game $G_{A,\Delta,\mu}$ (region graph)
- 2 Define a set of Bad states:

Bad is reachable \iff A is not Δ -DTA- μ -diagnosable

- 3 Solve safety game "Avoid Bad"
- 4 The most permissive winning strategy = set of all DTA- μ diagnosers

Synthesis for Codiagnosability

- 1 Construct n two-player games G^i (corresponding to G_{A,Δ,μ_i}^i)
- 2 Define the set of Bad_i states

Lemma

A is (Δ, \vec{D}) -codiagnosable iff there is a tuple of strategies \vec{f} s.t.

- (1) $\forall 1 \leq i \leq n$, $\vec{f}[i]$ is state-based on the game G^i , and
- (2) $\forall w \in \text{Tr}(A)$ $\left\{ \begin{array}{l} \text{If } S_i = \text{last}(\pi_{\Sigma_i}(w), f_i(G^i)), 1 \leq i \leq n, \\ \text{then } \exists 1 \leq j \leq n, \text{ s.t. } S_j \notin \text{Bad}_j \end{array} \right.$

Synthesis of a DTA Codiagnoser

Algorithm [Bouyer 2005]

- 1 Construct a **two-player game finite game** $G_{A,\Delta,\mu}$ (region graph)
- 2 Define a set of **Bad** states:
- 3 Solve **safety game** "Avoid Bad"
- 4 The **most permissive winning strategy** = set of all **DTA- μ diagnosers**

Synthesis for Codiagnosability

- 1 Construct n **two-player games** G^i (corresponding to G_{A,Δ,μ_i}^i)
- 2 Define the set of **Bad _{i}** states

Lemma

A is (Δ, \vec{D}) -codiagnosable iff there is a tuple of strategies \vec{f} s.t.

- (1) $\forall 1 \leq i \leq n, \vec{f}[i]$ is **state-based** on the game G^i , and
- (2) $\forall w \in \text{Tr}(A) \begin{cases} \text{If } S_i = \text{last}(\pi_{\Sigma_i}(w), f_i(G^i)), 1 \leq i \leq n, \\ \text{then } \exists 1 \leq j \leq n, \text{ s.t. } S_j \notin \text{Bad}_j \end{cases}$

Synthesis of a DTA Codiagnoser

Algorithm [Bouyer 2005]

- 1 Construct a **two-player game finite game** $G_{A,\Delta,\mu}$ (region graph)
- 2 Define a set of **Bad** states:
- 3 Solve **safety game** "Avoid Bad"
- 4 The **most permissive winning strategy** = set of all **DTA- μ diagnosers**

Synthesis for Codiagnosability

- 1 Construct n **two-player games** G^i (corresponding to G_{A,Δ,μ_i}^i)
- 2 Define the set of **Bad _{i}** states

Lemma

A is (Δ, \vec{D}) -codiagnosable iff there is a tuple of strategies \vec{f} s.t.

- (1) $\forall 1 \leq i \leq n$, $\vec{f}[i]$ is **state-based** on the game G^i , and
- (2) $\forall w \in \text{Tr}(A)$ $\left\{ \begin{array}{l} \text{If } S_i = \text{last}(\pi_{\Sigma_i}(w), f_i(G^i)), 1 \leq i \leq n, \\ \text{then } \exists 1 \leq j \leq n, \text{ s.t. } S_j \notin \text{Bad}_j \end{array} \right.$

Summary of the Results

Discrete Event Systems

Δ -codiagnosability	Codiagnosability	Optimal Delay	Synthesis
PSPACE-C.	PSPACE-C. (EXPTIME [†])	PSPACE	EXPTIME
PTIME*	PTIME	PTIME	EXPTIME

Timed Systems (Timed Automata)

Δ -codiagnosability	Codiagnosability	Optimal Delay	Synthesis
PSPACE-C.	PSPACE-C.	PSPACE	2EXPTIME-C.
PSPACE-C.	PSPACE-C.	PSPACE	2EXPTIME-C.

† Best known upper bound

* Corresponding diagnosis problems

References I

- [Sampath 1995] Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D.:
Diagnosability of discrete event systems.
IEEE Transactions on Automatic Control 40(9) (September 1995)
- [Jiang 2001] Jiang, S., Huang, Z., Chandra, V., Kumar, R.:
A polynomial algorithm for testing diagnosability of discrete event systems.
IEEE Transactions on Automatic Control 46(8) (August 2001)
- [Yoo 2002] Yoo, T.S., Lafortune, S.:
Polynomial-time verification of diagnosability of partially-observed discrete-event systems.
IEEE Transactions on Automatic Control 47(9) (September 2002) 1491-1495
- [Tripakis 2002] Tripakis, S.:
Fault diagnosis for timed automata.
In Damm, W., Olderog, E.R., eds.: Proceedings of the International Conference on Formal Techniques in Real Time and Fault Tolerant Systems (FTRTFT'02).
Volume 2469 of LNCS., Springer Verlag (2002) 205-224
- [Bouyer 2005] Bouyer, P., Chevalier, F., D'Souza, D.:
Fault diagnosis using timed automata.
In Sassone, V., ed.: Proceedings of the 8th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'05).
Volume 3441 of LNCS., Edinburgh, U.K., Springer Verlag (April 2005) 219-233

References II

- [Debouk 2000] Debouk, R., Lafortune, S., Teneketzis, D.:
Coordinated decentralized protocols for failure diagnosis of discrete event systems.
Discrete Event Dynamic Systems 10(1-2) (2000) 33-86
- [Wang 2007] Wang, Y., Yoo, T.S., Lafortune, S.:
Diagnosis of discrete event systems using decentralized architectures.
Discrete Event Dynamic Systems 17(2) (2007) 233-263
- [Kozen 1977] Kozen, D.:
Lower bounds for natural proof systems.
In: FOCS, IEEE (1977) 254-266
- [C. 2009] Cassez, F.:
A Note on Fault Diagnosis Algorithms.
In: 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, Shanghai, P.R. China, IEEE Computer Society (December 2009)

Proof of Theorem 6

Intersection Emptiness for DFA

Input: n deterministic finite automata $A_i, 1 \leq i \leq n$, over the alphabet Σ .

Problem: Check whether $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$.

Theorem ([Kozen 1977])

IE is PSPACE-hard even if in A_2, \dots, A_n all the states are accepting.

Proof of Theorem 6

Intersection Emptiness for DFA

Input: n deterministic finite automata $A_i, 1 \leq i \leq n$, over the alphabet Σ .

Problem: Check whether $\bigcap_{i=1}^n \mathcal{L}^*(A_i) = \emptyset$.

Reduction:

