First Steps Towards Incremental Diagnosis of Discrete-Event Systems

Alban Grastien[1], Marie-Odile Cordier[1], and Christine Largouët[2]

[1] Irisa, University of Rennes 1, Campus de Beaulieu, 35042 Rennes Cedex, France
{agrastie,cordier}@irisa.fr
[2] University of New Caledonia, BP. 4477, 98847 Nouméa Cedex, New Caledonia
largouet@univ-nc.nc

Abstract. This paper deals with the incremental off-line computation of diagnosis of discrete-event systems. Traditionally, the diagnosis is computed from the global automaton describing the observations emitted by the system on a whole time period. The idea of this paper is to slice this global automaton according to temporal windows and to compute local diagnoses for each of these windows. It is shown that, under some conditions, the global diagnosis can be computed from the local diagnosis. This paper presents the formalization used to compute an incremental diagnosis, relying on the new concept of *automata chain*. It is then shown that it is possible to take into account the diagnosis obtained for the previous temporal windows to incrementally compute the current diagnosis more efficiently. This work is a first and necessary step before considering the on-line diagnosis computation. The main difficulty is then to ensure the correct slicing of the observation automaton and to determine the appropriate temporal windows.

1 Introduction

It is well-established in the Model-Based Diagnosis community that a diagnosis is defined as the set of trajectories consistent with the observations. Different terminologies can be used as *histories* [1], *scenarios* [2], *narratives* [3], *consistent paths* [4] or *trajectories* [5]. A diagnosis is then formally defined as the synchronized product of the automaton modelling the system and the automaton modelling the observations emitted by the system on the considered time period. In an off-line context, this observation automaton can be huge (especially when taking into account uncertainties on delays) and its size depends directly on the length of the time period. This is for instance the case for the computation of an *a posteriori* diagnosis from observations collected on a few days, as for alarm logs in telecommunication networks. In this paper, we present the idea of slicing the observation automaton according to appropriate temporal windows in order to compute incrementally the diagnosis rather than globally considering the computation of diagnosis

To reach this objective, we propose the concept of automata chain to represent observations and diagnoses by slices. We then show that it is possible to

compute the global diagnosis from this modular representation of observations. For each temporal window, a local diagnosis is computed and the global one can be correctly represented by the automata chain of these local diagnoses. A first formalization of diagnosis by slices is given. The problem that appears is the huge size of these local diagnoses when computed in parallel. A second formalization is then proposed to compute incrementally the current diagnosis from the previous one, elaborated on a past temporal window. This work is a first and necessary step motivated by the ambitious problem of on-line incremental diagnosis.

The formalism used to represent the system model, the observations and the diagnosis is given in Section 2. Section 3 defines the automata chain concept and shows how the automata chain can be used to compute the global diagnosis. Two approaches for the incremental computation of the diagnosis are detailed and discussed in Section 4.

2 Automata and global diagnosis

This section deals with the problem of computing a global diagnosis without considering the problem of incrementality. These definitions are necessary for the incremental computing of diagnosis discussed in the following sections.

2.1 Automata and trajectories

The system considered evolves with the occurrence of *events* and an event can cause, by propagation, other events (case of reactive systems, see for example [6]). Consequently, events can occur simultaneously. We denote E the set of events.

The behaviours of the system are represented as classic automata:

Definition 1 (Automaton)
An automaton *is a tuple* $A = (Q, E, T, I, F)$ *where:*

- Q *is a set of states,*
- E *is a set of events,*
- $T \subseteq (Q \times 2^E \times Q)$ *is a set of transitions* $t = (q, l, q')$ *where* t *connects the source state* q *to the target state* q' *on a label* l *which is a non-empty set of events* $(l \subseteq E)$,
- $I \subseteq Q$ *is the set of initial states and*
- $F \subseteq Q$ *is the set of final states.*

We consider that the transition labels are non-empty sets of events. However, $\forall q \in Q$, $(q, \emptyset, q)$ is an implicit transition of T.

Definition 2 (Trajectory)
A trajectory*, denoted* $traj$ *on an automaton* $A = (Q, E, T, I, F)$ *is the couple of a* finite *state sequence* $(q_0, \ldots, q_n)$ *and of a label sequence* $(l_1, \ldots, l_n)$ *such that:*

- $\forall i \in \{0, \ldots, n\}, \ q_i \in Q$,
- $\forall i \in \{1, \ldots, n\}, \ t_i = (q_{i-1}, l_i, q_i) \in T$,
- $q_0 \in I$ *and*
- $q_n \in F$.

A trajectory is defined as a sequence of states (such that the first state is an initial state and the last state a final state) and a sequence of labels over the transitions between each state of the trajectory.

A trajectory can contain implicit transitions of the automaton. We consider that a trajectory is equal to the trajectory from which implicit transitions have been removed. Let $traj = ((q_0, \ldots, q_i, q_{i+1}, \ldots, q_n), (l_1, \ldots, l_i, l_{i+1}, \ldots, l_n))$ and $traj' = ((q_0, \ldots, q_i, q_i, q_{i+1} \ldots, q_n), (l_0, \ldots, l_i, \emptyset, l_{i+1}, \ldots, l_n))$. Then $traj = traj'$.

Two automata A and A' are identical if their set of trajectories are identical. We call *simplified automaton of* A the automaton $A' = A$ where all the states and transitions that do not appear in at least one trajectory have been removed. In the following, when computing new automata, only simplified ones are considered.

2.2 Synchronized automata

Definition 3 (Synchronization of labels)
Let l_1 be a label on E_1 and l_2 be a label on E_2. We say that l_1 and l_2 are synchronized iff $l_1 \cap (E_1 \cap E_2) = l_2 \cap (E_1 \cap E_2)$. Their synchronization, denoted $\Theta(l_1, l_2)$ is the label $l_1 \cup l_2$ on the set of events $E_1 \cup E_2$.

Two labels are synchronized if the synchronization events $(E_1 \cap E_2)$ present in one label are present in the other label.

Definition 4 (Synchronization)
Let $A_1 = (Q_1, E_1, T_1, I_1, F_1)$ and $A_2 = (Q_2, E_2, T_2, I_2, F_2)$ be two automata. The synchronized automaton *of A_1 and A_2, denoted $A_1 \otimes A_2$, is the automaton $A = (Q, E, T, I, F)$ defined by:*

- $Q = Q_1 \times Q_2$,
- $E = E_1 \cup E_2$,
- $T = \{((q_1, q_2), l, (q_1', q_2')) \mid \ \exists l_1, l_2,$
    - $(q_1 = q_1' \wedge l_1 = \emptyset) \vee (q_1, l_1, q_1') \in T_1$
    - $(q_2 = q_2' \wedge l_2 = \emptyset) \vee (q_2, l_2, q_2') \in T_2$
    - $l = \Theta(l_1, l_2)$
  $\}$,
- $I = I_1 \times I_2$ *and*
- $F = F_1 \times F_2$.

Each transition of the synchronized automaton A corresponds to a pair of transitions on automata A_1 and A_2 such that the labels of the transitions are synchronized.

It can be easily proved that $(A_1 \otimes A_2) \otimes A_3 = A_1 \otimes (A_2 \otimes A_3)$ with the following state renaming: $((q_1, q_2), q_3) \rightarrow (q_1, (q_2, q_3))$. In the following, and to simplify, we denote: $A = A_1 \otimes \ldots \otimes A_n = A_1 \otimes (\ldots \otimes A_n) = (A_1 \otimes \ldots) \otimes A_n = (Q, E, T, I, F)$ with $Q = Q_1 \times \ldots \times Q_n$. Moreover, we consider that $A_1 \otimes A_2 = A_2 \otimes A_1$. In a more general way we consider that $((q_1, q_2), q_3) = (q_1, q_2, q_3)$.

2.3 Diagnosis

Definition 5 (System model)
We denote by $MOD = (Q^{MOD}, E^{MOD}, T^{MOD}, I^{MOD}, F^{MOD})$ the system model. *I^{MOD} is the set of possible states at the time t_0. The final state may be any of the states: $F^{MOD} = Q^{MOD}$. The set of observable events is $E_{OBS}^{MOD} \subseteq E^{MOD}$.*

Let us now consider the observations and diagnosis definitions. Usually, due to uncertainties on the observations, we do not know the total order of the observations emitted by the system. Consequently the observations are represented by an automaton, each trajectory of which representing a possible order on the emitted observations during the period $[t_0, t_n]$.

Definition 6 (Observations)
The observations, *denoted OBS, are represented by an automaton describing the observable events emitted by the system during the period $[t_0, t_n]$.*

Definition 7 (Diagnosis)
The global diagnosis, *denoted Δ_n is an automaton describing the possible trajectories on the system model compatible with the observations emitted by the system during the period $[t_0, t_n]$.*

The global diagnosis of the system can be computed in the following way (see for instance [7, 8]):

$$\Delta_n = OBS_n \otimes MOD \tag{1}$$

3 Automata chain and global diagnosis

3.1 Objectives

At the end of the previous section we have presented a way to compute the global diagnosis of the system on the period $[t_0, t_n]$. Our goal is now to compute a diagnosis on the period $[t_0, t_i]$ $(i < n)$ and, given this diagnosis as well as the observations on the period $[t_i, t_{i+1}]$, to *incrementally* compute the diagnosis $[t_0, t_{i+1}]$. In order to achieve this goal we introduce the concept of *automata chain* and first apply it to the case of global diagnosis. We then extend the principle to the problem of incremental computation detailled Section 4.

Figure 1 illustrates the principle of slicing an automaton into an automata chain and conversely (called the reconstruction). The use of automata chain for

the diagnosis is also presented in this figure. Given the observation automaton OBS_n and the model of the system MOD, it is possible to compute the global diagnosis by synchronization, as presented Section 2. The idea is to *slice* the observations automaton into a sequence of automata OBS^i called *automata chain*, so that the original observation automaton can be rebuilt from the automata chain by a *reconstruction*. Each automaton OBS^i is *local to a temporal window* $\mathcal{W}^i$. The local diagnosis of the temporal window $\mathcal{W}^i$ is computed using OBS^i. Finally, the global diagnosis is obtained by reconstruction of the local diagnoses.
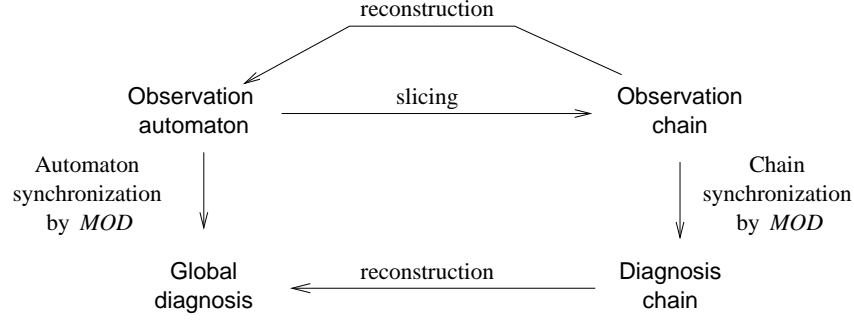

Fig. 1. Principle of the use of an automata chain

We define below the automata chain and give the properties that enable us to compute the diagnosis with the automata chain as illustrated on Figure 1.

3.2 Υ-transition

We define a special class of events called Υ-events (denoted Υ_i for $i \in \mathbf{N}$). An Υ-event corresponds to a clock tick associated with the date t_i. It does not correspond to an event of the system.

We assume that an Υ-event and a system event $e \in E$ cannot occur simultaneously: if l is such that $\exists k$, $\Upsilon_k \in l$ then $l = \{\Upsilon_k\}$. In the following of this paper, we note $l = \Upsilon_k$ for $l = \{\Upsilon_k\}$. We call Υ-transition a transition labeled by Υ_k.

We change the definition of synchronization by adding the following property to the definition 3. To be synchronized, two labels l_1 and l_2 should satisfy this condition: if $(\exists i \in \{1, 2\}, j \in \{1, 2\}, j \neq i)$, $l_i = \Upsilon_k$, then either $l_j = l_i$ or $l_j = \emptyset$. Given this new definition, we ensure that the property on the Υ-event is satisfied by the synchronization of two labels.

A temporal window $\mathcal{W}^i$ is defined as the period between two ticks represented by Υ_{i-1} and Υ_i. The period $\mathcal{W}^i$ is the period coming after Υ_1 and the period $\mathcal{W}^n$ is the period coming after the last tick Υ_{n-1}.

3.3 Automata chain

Definition 8 (Automata chain)
A sequence of automata $(A^1, \ldots, A^n)$ with $A^i = (Q^i, E^i, T^i, I^i, F^i)$ is called an automata chain and denoted $\mathcal{E}_A$ if $\forall i \in \{1, \ldots, n\}, \forall k \in \mathbf{N}, \; \Upsilon_k \notin E^i$.

The meaning of an *automata chain* is the following: from a state q_1 from I^1, it is possible to reach by the transitions of A^1 a state q of $F^1 \cap I^2$; and then, it is possible to visit A^2 from q, etc. The automaton A^i is associated to the temporal window $\mathcal{W}^i$. An automata chain $(A^1, \ldots, A^n)$ can also be represented $((A^1, \ldots, A^{n-1}), A^n)$.

The length of the chain is the number of automata in the chain. A 3-long automata chain is presented on Figure 2. To simplify the representation the labels over the transitions are not represented.
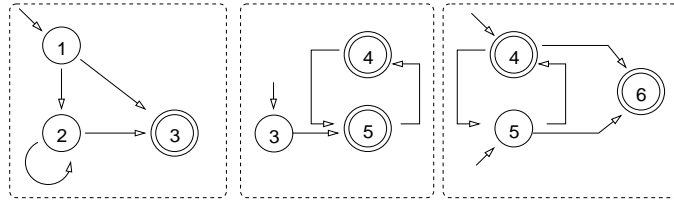

Fig. 2. Chain of three automata

Definition 9 (Chain concatenation)
Let $\mathcal{E}_A = (A^1, \ldots, A^n)$ be an automata chain with $A^i = (Q^i, E^i, T^i, I^i, F^i)$. The chain concatenation of $\mathcal{E}_A$, denoted $\oplus \mathcal{E}_A$ is an automaton $A' = (Q', E', T', I', F')$ defined by:

- $Q' = (Q^1 \cup \ldots \cup Q^n) \times \{\mathcal{W}^1, \ldots, \mathcal{W}^n\}$,
- $E' = (E^1 \cup \ldots \cup E^n) \cup \{\Upsilon_1, \ldots, \Upsilon_{n-1}\}$,
- $T' = \{((q, \mathcal{W}^i), l, (q', \mathcal{W}^i)) \mid (q, l, q') \in T^i\} \; \cup$
  $\{((q, \mathcal{W}^i), \Upsilon_i, (q, \mathcal{W}^{i+1})) \mid \; q \in F^i \wedge q \in I^{i+1}\}$,
- $I' = I^1 \times \{\mathcal{W}^1\}$ *and*
- $F' = F^n \times \{\mathcal{W}^n\}$.

Since different automata of an automata chain can have states in common, the knowledge of a state q is not necessary related to a single temporal window. Consequently, the states q of the automata chain are said *relative* while the states $(q, \mathcal{W}^i)$ from the chain concatenation are said *absolute*. The concatenation transforms relative states into absolute states. The automaton obtained by a concatenation is called an *absolute automaton*.

The concatenation of the automata chain presented Figure 2 is shown Figure 3. To simplify, the states $(q, \mathcal{W}^i)$ are noted (q, i).
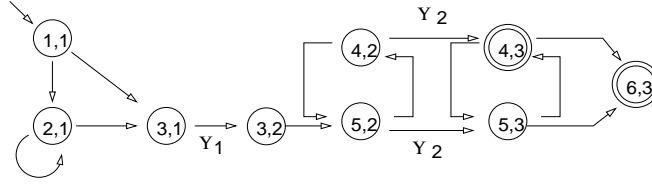
Fig. 3. Concatenation of the chain of Figure 2

Generally, we note: $\oplus \mathcal{E}_A = A^1 \oplus \ldots \oplus A^n = (A^1 \oplus \ldots \oplus A^{n-1}) \oplus A^n$.

We introduce now the representation of an automaton by an automata chain. In order to achieve this goal, we first give some definitions about trajectories.

Definition 10 (Trajectory abstraction)
Let $traj' = ((q'_0, \ldots, q'_n), (l'_1, \ldots, l'_n))$ be a trajectory on absolute states. Then, the abstraction of $traj'$ is the trajectory defined by $traj = ((q_0, \ldots, q_n), (l_1, \ldots, l_n))$ so that:

- *$\forall i \in \{0, \ldots, n\}$, $\exists k$, $q'_i = (q_i, \mathcal{W}^k)$ and*
- *$\forall i \in \{1, \ldots, n\}$, $(\exists k, l'_i = \Upsilon_k \Rightarrow l_i = \emptyset) \wedge (\nexists k, l'_i = \Upsilon_k \Rightarrow l_i = l'_i)$.*

Definition 11 (Automaton abstraction)
Let $A' = (Q', E', T', I', F')$ be an automaton on absolute states and let $A = (Q, E, T, I, F)$ be an automaton on relative states. A is an abstraction of A', denoted $A \simeq_{abs} A'$, iff;

- *for any trajectory $traj$ of A, there exists $traj'$ from A' and a trajectory $traj_2$ from A so that $traj_2 = traj$ and $traj_2$ is the abstraction of $traj'$ and*
- *for any trajectory $traj'$ from A', there exists a trajectory $traj$ from A so that $traj$ is the abstraction of $traj'$.*

Definition 12 (Reconstruction)
Let $\mathcal{E}_A$ be an automata chain. Let $A = (Q, E, T, I, F)$ an automaton so that $\forall k$, $\Upsilon_k \notin E$. A is the reconstruction of $\mathcal{E}_A$ (denoted $A \simeq_{rec} \mathcal{E}_A$) iff A is an abstraction of $\oplus \mathcal{E}_A$.
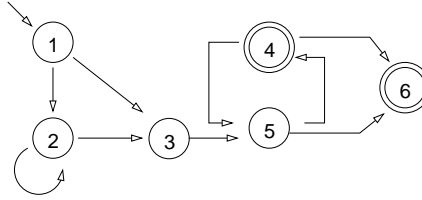
Definition 13 (Slicing)
Let A be an automaton and $\mathcal{E}_A$ be an automata chain. $\mathcal{E}_A$ is a slicing of A iff A is a reconstruction of $\mathcal{E}_A$.
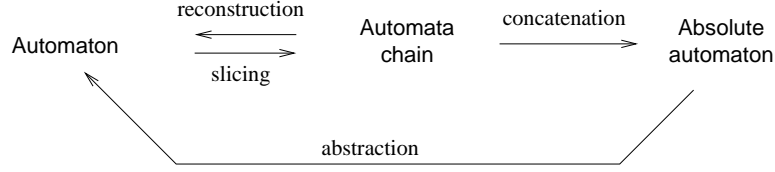
Fig. 4. Links between the automaton, the automata chain and the absolute automaton

The links between the automaton, the automata chain and the absolute automaton are presented on Figure 4.

The automaton presented Figure 5 is the abstraction of the automaton shown Figure 3. Thus, the automata chain given Figure 2 is a slicing of this abstract automaton (cf. Figure 5).

Fig. 5. Abstraction of the automaton of Figure 3

Property 1 *Let A be an automaton. Let $\mathcal{E}_A$ be a slicing of A. Let $A_2 = (Q_2, E_2, T_2, I_2, F_2)$ be an automaton so that $\forall k$, $\Upsilon_k \notin E_2$. Then $A \otimes A_2 \simeq_{abs} (\oplus \mathcal{E}_A) \otimes A_2$.*

3.4 Synchronization of automata chain

Definition 14 (Prefix-closed automaton)
Let $A = (Q, E, T, I, F)$ be an automaton. The prefix-closed automaton *of A, denoted A^+, is equal to the automaton A in which all states are final ($F^+ = Q$).*

Definition 15 (Suffix-closed automaton)
Let $A = (Q, E, T, I, F)$ be an automaton. The suffix-closed automaton *of A, denoted A^-, is equal to the automaton A in which all states are initial ($I^- = Q$).*

We denote $A^{\#}$ the automaton prefix-closed suffix-closed ($A^{\#} = (A^{+})^{-} = (A^{-})^{+}$).

Definition 16 (Synchronization of a chain by an automaton)
Let $\mathcal{E}_A = (A^1, \ldots, A^n)$ be an automata chain. Let A_2 be an automaton. The synchronization of $\mathcal{E}_A$ by A_2 is the automata chain, denoted $\mathcal{E}_A \otimes A_2$, defined by: $\mathcal{E}_A \otimes A_2 = (A^1 \otimes A_2^+, A^2 \otimes A_2^{\#}, \ldots, A^{n-1} \otimes A_2^{\#}, A^n \otimes A_2^-)$.

The synchronisation of a chain automata with an automaton A_2 consists in synchronizing each automaton of the chain with the automaton A_2. In the synchronization of an automata chain $\mathcal{E}_A = (A^1, \ldots, A^n)$ with an automaton $A_2 = (Q_2, E_2, T_2, I_2, F_2)$, all the states of A_2 are considered to be initial, during the synchronization with A^i ($i \neq 1$), since the state q_2 of A_2 current during Υ_{i-1} is not necessary initial. Thus, we have used the suffix-closed automaton of A_2. A similar reasoning has been applied for the final state and then the prefix-closed automaton is also used.

Property 2
Let $\mathcal{E}_A$ be an automata chain and let $A_2 = (Q_2, E_2, T_2, I_2, F_2)$ be an automaton so that $\Upsilon \notin E_2$. Then, $\oplus(\mathcal{E}_A \otimes A_2) = (\oplus\mathcal{E}_A) \otimes A_2$.

Proof. This property can be checked by proving that the set of trajectories of $\oplus(\mathcal{E}_A \otimes A_2)$ and the set of trajectories of $(\oplus\mathcal{E}_A) \otimes A_2$ are equal.

Corollary 1
Let A be an automaton. Let $\mathcal{E}_A$ be a slicing of A. Let $A_2 = (Q_2, E_2, T_2, I_2, F_2)$ be an automaton so that $\Upsilon \notin E_2$. Then, $\mathcal{E}_A \oplus A_2$ is a slicing of $A \otimes A_2$.

3.5   Diagnosis

Let $MOD = (Q^{MOD}, E^{MOD}, T^{MOD}, I^{MOD}, F^{MOD})$ be the model of the system as presented in Section 2. Let us remark that MOD does not give any information on the final states of the system ($F^{MOD} = Q^{MOD}$). Thus, we have the following properties: $MOD^+ = MOD$ and $MOD^{\#} = MOD^-$.

Let OBS_n be the automaton representing the observations emitted by the system and $\mathcal{E}_A$ be a slicing of OBS_n.

We have:
$$\Delta_n \simeq_{rec} \mathcal{E}_{OBS_n} \otimes MOD \tag{2}$$

4   Incremental diagnosis

In this section, we consider the incremental computation of the diagnosis. Our goal is to compute the diagnosis on the temporal window $\mathcal{W}_{i+1} = [t_0, t_{i+1}]$ from the diagnosis on the window $\mathcal{W}_i$ and the observations on the window $\mathcal{W}^{i+1}$.

In the following, the diagnosis is formalized as a diagnosis chain. We first present the interest of using diagnosis chain rather than the complete diagnosis

automata as described previously. We first propose a parallelized computing to elaborate the diagnosis from the results obtained in the previous section. Finally, we show that it is possible to take into account the diagnosis obtained for the previous temporal windows to incrementally compute the current diagnosis in a more efficient way.

4.1 Diagnosis chain

As presented Section 2, the diagnosis is defined as a set of trajectories on the system model and then can be easily formalized by an automata chain.

Definition 17 (Trajectories concatenation)
Let $traj^k = ((q_0^k, \ldots, q_{nk}^k), (l_1, \ldots, l_{nk}))$ be i trajectories so that $\forall k \in \{1, \ldots, i-1\}$, $q_{nk}^k = q_0^{k+1}$. Then, the concatenation *of the i trajectories $traj^k$ is defined by: $traj = ((q_0^1, \ldots, q_{n1}^1, q_1^2, \ldots, q_1^i, \ldots, q_{ni}^i), (l_1^1, \ldots, l_{n1}^1, \ldots, l_1^i, \ldots, l_{ni}^i))$.*

Property 3
Let A be an automaton and let $\mathcal{E}_A = (A^1, \ldots, A^i)$ be a slicing of A.

- *Let $\forall k \in \{1, \ldots, i\}$, $traj^k = ((q_0^k, \ldots, q_{n(k)}^k), (l_1, \ldots, l_{n(k)}))$, i trajectories on the automata A^k so that $traj$, the concatenation of the i trajectories, exists. Then, $traj$ is a trajectory of A.*
- *Let $traj$ be a trajectory of A. Then, there exists i trajectories $traj^k$ on the automata A^k so that the concatenation of the i trajectories is $traj$.*

This property is a logical extension of the previous properties. It shows that an automata chain *represents* the set of trajectories of the automaton that is obtained by reconstructing the chain.

4.2 Parallelized computation

Let $\mathcal{E}_{\Delta i} = (\Delta^1, \ldots, \Delta^i)$ be the diagnosis of the period $\mathcal{W}_i$ incrementally computed. Let $\mathcal{E}_{\Delta(i+1)}$ be the diagnosis of the system during the window $\mathcal{W}_{i+1}$. Then, we have:

$$\mathcal{E}_{\Delta i+1} = (\Delta^1, \ldots, \Delta^i, \Delta^{i+1}) \quad \text{with } \Delta^{i+1} = OBS^{i+1} \otimes MOD^- \tag{3}$$

This result comes from the fact that $MOD^{\#} = MOD^-$.

The advantage of this approach is that the local diagnoses can be computed in parallel. However, if the set of states Q^{MOD} is huge, then MOD^- has a huge number of initial states, and also Δ^{i+1}. The computation of Δ^{i+1} can then be very expensive. It is thus necessary to limit as much as possible the set of initial states in the diagnosis on a temporal window $\mathcal{W}^i$. This is the goal of the second approach uses the *incremental synchronization.*

4.3 Incremental synchronization

Definition 18 (Restriction)
Let $A = (Q, E, T, I, F)$ be an automaton. The restriction *of the automaton A by the set I', denoted $A[I']$, is the automaton $A' = (Q, E, T, I \cap I', F)$.*

Definition 19 (Incremental synchronization)
Let $\mathcal{E}_A = (A^1, \ldots, A^n)$ be an automata chain. Let A_2 be an automaton. The incremental synchronization *of the automata chain $\mathcal{E}_A$ and A_2 is the automata chain $\mathcal{E}_B$, denoted $\mathcal{E}_A \odot A_2$ and defined by $\mathcal{E}_B = (A_B^1, \ldots, A_B^n)$ with $\forall i \in \{1, \ldots, n\}$ so that:*

- $A_B^1 = A^1 \otimes A_2^+$,
- $\forall i \in \{2, \ldots, n-1\}$, $A_B^i = (A^i \otimes A_2^{\#})[F_B^{i-1}]$ *and*
- $A_B^n = (A^n \otimes A_2^-)[F_B^{n-1}]$.

The incremental synchronization restricts the automaton A_B^i by the set of final states of the previous automaton A_B^{i-1}.

Property 4
Let $\mathcal{E}_A$ be an automata chain and $A_2 = (Q_2, E_2, T_2, I_2, F_2)$ so that $\Upsilon \notin E_2$. Then, $\oplus(\mathcal{E}_A \odot A_2) = \oplus(\mathcal{E}_A \otimes A_2)$.

We note $\Delta^i = (Q_\Delta^i, E_\Delta^i, T_\Delta^i, I_\Delta^i, F_\Delta^i)$. Let $\mathcal{E}_{\Delta_i} = (\Delta^1, \ldots, \Delta^i)$ be the diagnosis of the system during the temporal window $\mathcal{W}_i$. It is possible to incrementally compute the diagnosis Δ^{i+1} on the temporal window $\mathcal{W}_{i+1}$:

$$\mathcal{E}_{\Delta i+1} = (\Delta^1, \ldots, \Delta^i, \Delta^{i+1}) \quad \text{with } \Delta^{i+1} = (OBS^{i+1} \otimes MOD^-)[F_\Delta^i] \qquad (4)$$

This method enables us to limit the number of initial states of the local diagnoses.

5 Conclusion

In this paper, we have shown that an automaton can be sliced and represented by an automata chain. We explain the relations between these two representations and define the synchronization operation for each of them. Diagnosis is usually formally defined as resulting from synchronization between the system automaton and the observation automaton. We propose to replace the observation automaton by an automata chain, each automaton representing the observation emitted during the corresponding temporal window. It is then possible to compute the diagnosis as an automata chain, each automaton representing a local diagnosis for a given temporal window. We show that it is equivalent to compute the global diagnosis for the global observation automaton. Moreover, we show that this computation can be performed in an incremental way, and thus

more efficiently by taking profit of the already computed diagnoses to improve the computation of the current one.

This work concerns the off-line case. The next step is to study the on-line case. The incremental approach means computing a diagnosis for a given time, and then extending it by taking into account the next temporal window. It requires to be able to build, on-line and incrementally, the observation automata chain. The main difficulties are to determine the adequate temporal windows and to ensure that the slicing is correct whenever the future observations are unknown.

References

1. Baroni, P., Lamperti, G., Pogliano, P., Zanella, M.: Diagnosis of large active systems. Artificial Intelligence **110** (1999) 135–183
2. Cordier, M.O., Thiébaux, S.: Event-based diagnosis for evolutive systems. In: 5th International Workshop on Principles of Diagnosis (DX-94). (1994) 64–69
3. Barral, C., McIlraith, S., Son, T.: Formulating diagnostic problem solving using an action language with narratives and sensing. In: International Conference on Knowledge Representation and Reasoning (KR'2000). (2000) 311–322
4. Console, L., Picardi, C., Ribaudo, M.: Diagnosis and diagnosability analysis using PEPA. In: 14th European Conference on Artificial Intelligence (ECAI-00), Berlin, Allemagne (2000) 131–135
5. Cordier, M.O., Largouët, C.: Using model-checking techniques for diagnosing discrete-event systems. In: 12th International Workshop on Principles of Diagnosis (DX-2001). (2001) 39–46
6. Lamperti, G., Zanella, M.: Diagnosis of Active Systems. Kluwer Academic Publishers (2003)
7. Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D.: Failure diagnosis using discrete-event models. In: IEEE Transactions on Control Systems Technology (CST-96). (1996) 105–124
8. Rozé, L., Cordier, M.O.: Diagnosing discrete-event systems : extending the "diagnoser approach" to deal with telecommunication networks. Journal on Discrete-Event Dynamic Systems: Theory and Applications (JDEDS) **12 (1)** (2002) 43–81 errata **14 (1)** (2004) 131.