Notes on the
Fault Diagnosis and Diagnosability of Discrete Event Systems

Alessandro Giua
Aix-Marseille University, Marseille, France & University of Cagliari, Italy
Email: giua@diee.unica.it

version date: 13 April 2017

This document contain a short introduction to the seminal approach to the diagnosis of automata developed by Lafortune and coworkers [2]. A more comprehensive presentation of the approach can be found in [1].

1 The plant model

The system to be diagnosed is modeled as a DFA. Since we are not interested in the set of final states, we will denote such an automaton by $G = (X, E, \delta, x_0)$. The behavior of the system is described by the prefix-closed language $L(G)$ generated by G.

The DFA G models both the normal and the faulty behavior. Its alphabet can be partitioned as $E = E_o \cup E_{uo}$ where:

- E_o: is the set of *observable events*;
- E_{uo}: is the set of *unobservable events*. The set of unobservable events can be further partitioned as $E_{uo} = E_f \cup E_{reg}$ where
  - E_f is the set of *fault events*[1] ;
  - E_{reg} is the set of *regular events* that, although not observable, do not describe a faulty behavior.

In the rest of the chapter the following assumptions hold.

(A1) The DFA G does not contain dead states.

(A2) The DFA G does not contain cycles of unobservable events.

Assumption (A1) is made for the sake of simplicity. On the contrary, assumption (A2) is necessary and ensures that the system G does not generate sequences of unobservable events whose length can be infinite.

[1]The set of fault events may also partitioned into m disjoint subsets that represent different of fault classes: $E_f = E_{f,1} \cup E_{f,2} \cup \ldots \cup E_{f,r}$. However, in the rest of this section we will consider a single fault class for sake of simplicity.

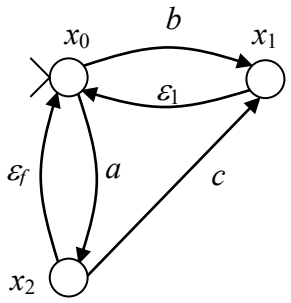

Figure 1: A DFA G with set of observable events $E_o = \{a, b, c\}$, set of unobservable regular events $E_{reg} = \{\varepsilon_1\}$ and set of unobservable fault events $E_{reg} = \{\varepsilon_f\}$.

Example 1 Consider the automaton in Figure 1. The set of observable events is $E_o = \{a, b, c\}$ while the set of unobservable events is $E_{uo} = \{\varepsilon_1, \varepsilon_f\}$. In particular the set of regular events is $E_{reg} = \{\varepsilon_1\}$ and the set of fault events is $E_{reg} = \{\varepsilon_f\}$. The automaton satisfies both Assumption A1 and A2. $\diamond$

Let us define the projection operator on the set of observable events.

Definition 1 Given a DFA G with alphabet $E = E_o \cup E_{uo}$, the *projection operator* on the set of observable events is denoted by $P : E^* \to E_o^*$ and is defined as

$$\begin{cases} P(\varepsilon) = \varepsilon & \\ P(e) = e, & \text{if } e \in E_o\,; \\ P(e) = \varepsilon, & \text{if } e \in E_{uo}\,; \\ P(se) = P(s)P(e), & s \in E^*, e \in E\,. \end{cases}$$

The *inverse projection operator*[2] with codomain in $L(G)$ is denoted by $P^{-1} : E_o^* \to 2^{L(G)}$ and is defined as

$$P^{-1}(w) = \{s \in L(G) \mid P(s) = w\}.$$

▲

Thus, the projection operator P simply "erases" the unobservable events in a string, while the inverse projection associates to a sequence w of observable events the set of strings in the language of G whose projection is w. In the rest of this section we will denote by $s \in E^*$ a string of events generated by the DFA and by $w \in E_o^*$ an observed word, i.e., the observable projection of a generated string.

Assume that a DFA, starting from the initial state, generates a string $s \in E^*$ thus reaching a new state $x = \delta^*(x_0, s)$. Due to the projection mask, an external agent observes a word $w = P(s) \in E_o^*$, as shown in Figure 2. In general however the external agent may not be able to detect the exact string that has produced this observation or the exact state that has been reached.

Definition 2 Given a DFA $G = (X, E, \delta, x_0)$ with alphabet $E = E_o \cup E_{uo}$, for each word $w \in E_o^*$ we define:

[2]Properly speaking we should denote this operator by $P^{-1}_{L(G)}$ but the subscript will be omitted to avoid a cumbersome notation.

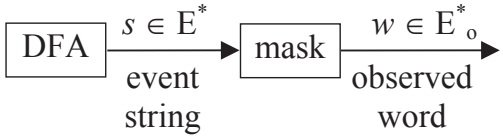

Figure 2: Observation of a DFA through a projection mask.

- $\mathcal{S}(w) = P^{-1}(w) \subseteq L(G)$ the set of *strings consistent with observation* w, i.e., the set of strings in the language of G that produce the observation w;
- $\mathcal{X}(w) = \{x \in X \mid (\exists s \in \mathcal{S}(w))\ \delta^*(x_0, s) = x\}$ the set of *states consistent with observation* w, i.e., the set of states in which G may be after w has been observed. ▲

Example 2 Consider the automaton in Figure 1 where the set of observable events is $E_o = \{a, b, c\}$.

Assume word bb is observed. Two different evolutions may have produced this observation:

$$x_0 \xrightarrow{b} x_1 \xrightarrow{\varepsilon_1} x_0 \xrightarrow{b} x_1$$
$$x_0 \xrightarrow{b} x_1 \xrightarrow{\varepsilon_1} x_0 \xrightarrow{b} x_1 \xrightarrow{\varepsilon_1} x_0$$

Hence for this observation the set of consistent strings is $\mathcal{S}(bb) = \{b\varepsilon_1 b,\ b\varepsilon_1 b\varepsilon_1\}$ while the set of consistent states is $\mathcal{X}(bb) = \{x_0, x_1\}$.

Consider word $bc \in E_o^*$. Since no string generated by the plant can produces this observation it holds $\mathcal{S}(bc) = \mathcal{X}(bc) = \emptyset$. ◇

An additional notation we will use if the following.

Definition 3 Given a string $s \in E$, the *support* of s is

$$||s|| = \{e \in E \mid |s|_e > 0\} \subseteq E,$$

and consists of the set of events that appear at least once in the string. ▲

Example 3 Consider again the automaton in Figure 1 whose alphabet is $E = \{a, b, c, \varepsilon_1, \varepsilon_f\}$. The support of string $s = a\varepsilon_f ac \in E^*$ is $||s|| = \{a, c, \varepsilon_f\}$. ◇

2 Diagnosis

In a fault diagnosis problem we want to determine, based on the observed word $w \in E_o^*$, if a fault has occurred, i.e., if a transition labeled with a symbol in E_f has fired. This leads to the definition a diagnosis problem.

Problem 1 *Given a DFA G with alphabet $E = E_o \cup E_{uo}$ and set of fault events $E_f \subseteq E_{uo}$ and given an observed word $w \in E_o^*$, the* diagnosis problem *consists in determining if a fault has occurred, i.e., if an evolution containing a transition with a label in E_f has produced the observation w.*

Solving a diagnosis problem requires constructing a diagnosis function.

Definition 4 Given a DFA G with alphabet $E = E_o \cup E_{uo}$ and set of fault events $E_f \subseteq E_{uo}$, a *diagnosis function*

$$\varphi : E_o^* \to \{N, F, U\}$$

associates to each observed word $w \in E_o^*$ a diagnosis state $\varphi(w) \in \{N, F, U\}$ as follows.

- $\varphi(w) = N$ (no fault): if for all $s \in P^{-1}(w)$ it holds $||s|| \cap E_f = \emptyset$. In such a case no string s consistent with the observed word w contains a fault event, hence no fault has occurred.
- $\varphi(w) = F$ (fault): if for all $s \in P^{-1}(w)$ it holds $||s|| \cap E_f \neq \emptyset$. In such a case all strings s consistent with the observed word w contain a fault event, hence a fault has certainly occurred.
- $\varphi(w) = U$ (uncertain): if there exist $s', s'' \in P^{-1}(w)$ such that $||s'|| \cap E_f = \emptyset$ and $||s''|| \cap E_f \neq \emptyset$. In such a case there exists two strings s' and s'' consistent with the observed word w, one containing a fault event and one not containing a fault event. Hence a fault may or may not have occurred. ▲

We remarks that when different fault classes $E_{f,1}, E_{f,2}, \ldots, E_{f,r}$ are given, one wants to diagnose separately each class i determining if a fault in this class has occurred, i.e., if a transition labeled with a symbol in $E_{f,i}$ has fired. This can be done solving r diagnosis problems, i.e., constructing r diagnosis functions φ_i, for $i = 1, 2, \ldots, r$. However, this case will not be discussed.

Example 4 Consider the automaton in Figure 1 where the set of observable events is $E_o = \{a, b, c\}$ and the set of fault events is $E_{reg} = \{\varepsilon_f\}$. The diagnosis function for this DFA is partially described in the following table where we have also listed for each observed word w the set of consistent strings $\mathcal{S}(w)$ and the set of consistent states $\mathcal{X}(w)$.

w	$\mathcal{S}(w) = P^{-1}(w)$	$\mathcal{X}(w)$	$\varphi(w)$
ε	ε	$\{x_0\}$	N
a	$\{\, a,\ a\varepsilon_f \,\}$	$\{x_0, x_2\}$	U
b	$\{\, b,\ b\varepsilon_1 \,\}$	$\{x_0, x_1\}$	N
aa	$\{\, a\varepsilon_f a,\ a\varepsilon_f a\varepsilon_f \,\}$	$\{x_0, x_2\}$	F
$\vdots$	$\vdots$	$\vdots$	$\vdots$

◇

A more interesting way of representing a diagnosis function is by means of a *diagnoser*, i.e., a DFA on the alphabet of observable events.

Definition 5 A *diagnoser* for DFA $G = (X, E, \delta, x_0)$ with alphabet $E = E_o \cup E_{uo}$ and set of fault events $E_f \subseteq E_{uo}$ is a DFA

$$Diag(G) = (Y, E_o, \delta_y, y_0)$$

on alphabet E_o such that

- $Y \subseteq 2^{X \times \{N,F\}}$, i.e., each state of the diagnoser is a set of pairs

  $$y = \{(x_1, \gamma_1), (x_2, \gamma_2), \ldots, (x_k, \gamma_k)\},$$

  where $x_i \in X$ and $\gamma_i \in \{N, F\}$, for $i = 1, 2, \ldots, k$.

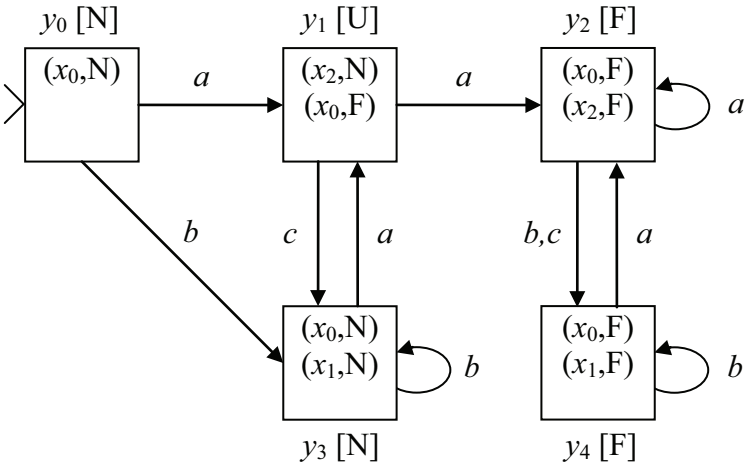

Figure 3: Diagnoser automaton $Diag(G)$ for the DFA G in Figure 1.

- $\delta_y^*(y_0, w) = y_w$ if and only if

$$\begin{aligned} y_w \;=\; & \{(x,N) \mid (\exists s \in \mathcal{S}(w))\ \delta^*(x_0,s) = x, ||s|| \cap E_f = \emptyset\} \\ & \cup \{(x,F) \mid (\exists s \in \mathcal{S}(w))\ \delta^*(x_0,s) = x, ||s|| \cup E_f \neq \emptyset\}, \end{aligned}$$

  i.e., in $Diag(G)$ starting from y_0 word w yields a state y_w containing:

  (a) all pairs (x, N) where x can be reached in G executing a string consistent with w that does not contain a fault event;

  (b) all pairs (x, F) where x can be reached in G executing a string consistent with w that contains a fault event.

To each state $y = \{(x_1, \gamma_1), (x_2, \gamma_2), \ldots (x_k, \gamma_k)\}$ of $Diag(G)$ we associate a diagnosis value $\varphi(y)$ such that:

- $\varphi(y) = N$ (no fault state): if $\gamma_i = N$ for all $i = 1, 2, \ldots, k$;
- $\varphi(y) = F$ (fault state): if $\gamma_i = F$ for all $i = 1, 2, \ldots, k$;
- $\varphi(y) = U$ (uncertain state): if there exist $i, j \in \{1, 2, \ldots, k\}$ such that $\gamma_i = N$ and $\gamma_j = F$. ▲

Thus a diagnoser allows one to associate to each observed work w a diagnosis state $\varphi(w) = \varphi(y_w)$ where $y_w = \delta_y^*(y_0, w)$ is the state reached in $Diag(G)$ by executing word w. Furthermore, the diagnoser also contains the information on the set of states consistent with w, because $\mathcal{X}(w) = \{x \in X \mid y_w = \delta_y^*(y_0, w), (x, \gamma) \in y_w\}$.

Example 5 Consider the plant in Figure 1 where the set of observable events is $E_o = \{a, b, c\}$ and the set of fault events is $E_f = \{\varepsilon_f\}$. The diagnoser for this DFA is shown in Figure 3, where we have labeled each state y of $Diag(G)$ with its corresponding diagnosis value $\varphi(y)$ in square brackets. ◇

A formal algorithm for constructing the diagnoser of a plant G is now given. This algorithm is similar to the algorithm used to compute the observer of a NFA G (i.e., the DFA equivalent to G). However, we now need to keep track not only of the possible states in which the plant can be but whether these states can be reached with or without firing a fault transition.

Algorithm 1 Construction of a diagnoser.
Input: A DFA $G = (X, E, \delta, x_0)$ with $E = E_o \cup E_{uo} = E_o \cup E_{reg} \cup E_f$
Output: A Diagnoser $Diag(G) = (Y, E_o, \delta_y, y_0)$ with $L(Diag(G)) = P(L(G))$.

1. **For all** states $x \in X$ of G compute the set

$$D_{reg}(x) = \{\bar{x} \in X \mid (\exists s \in E^*_{reg})\ \delta^*(x, s) = \bar{x})\}$$

containing all states reachable from x executing a (possibly empty) sequence of regular unobservable transitions and the set

$$D_f(x) = \{\bar{x} \in X \mid (\exists s \in E^*_{uo} \setminus E^*_{reg})\ \delta^*(x, s) = \bar{x}\}$$

containing all states reachable from x executing a sequence of unobservable transitions that contain at least one fault. Note that by definition $x \in D_{reg}(x)$. Also note that it may happen that $D_{reg}(x) \cap D_f(x) \neq \emptyset$, since a state $\bar{x}$ may be reachable from x by two different sequence of unobservable transitions, one that does not contain a fault, and one that contains a fault.

2. **Let**

$$y_0 = \{(x, N) \mid x \in D_{reg}(x_0)\} \cup \{(x, F) \mid x \in D_f(x_0)\},$$

i.e., the initial state of $Diag(G)$ is a set of pairs (x, γ) where:

- $\gamma = N$ (no fault) if x is reachable from x_0 executing a sequence of unobservable transitions that does not contain a fault;
- $\gamma = F$ (fault) if x is reachable from x_0 executing a sequence of unobservable transitions that contains a fault.

3. **Let** $Y = \emptyset$ and $Y_{new} = \{y_0\}$.

(At the end of the algorithm Y will contain all states of $Diag(G)$, while the set Y_{new} contains at each step the states of $Diag(G)$ still to be explored.)

4. Select a state $y \in Y_{new}$.

   (a) **For all** $e \in E_o$:

      i. Define the sets:

$$\alpha(y, e) = \{(x', \gamma) \mid (x, \gamma) \in y, x' = \delta(x, e)\}$$

and

$$\beta_1(y, e) = \{(\bar{x}'', N) \mid (x', N) \in \alpha(y, e), x'' \in D_{reg}(x')\},$$
$$\beta_2(y, e) = \{(\bar{x}'', F) \mid (x', N) \in \alpha(y, e), x'' \in D_f(x')\},$$
$$\beta_3(y, e) = \{(\bar{x}'', F) \mid (x', F) \in \alpha(y, e), x'' \in D_{reg}(x') \cup D_f(x')\}.$$

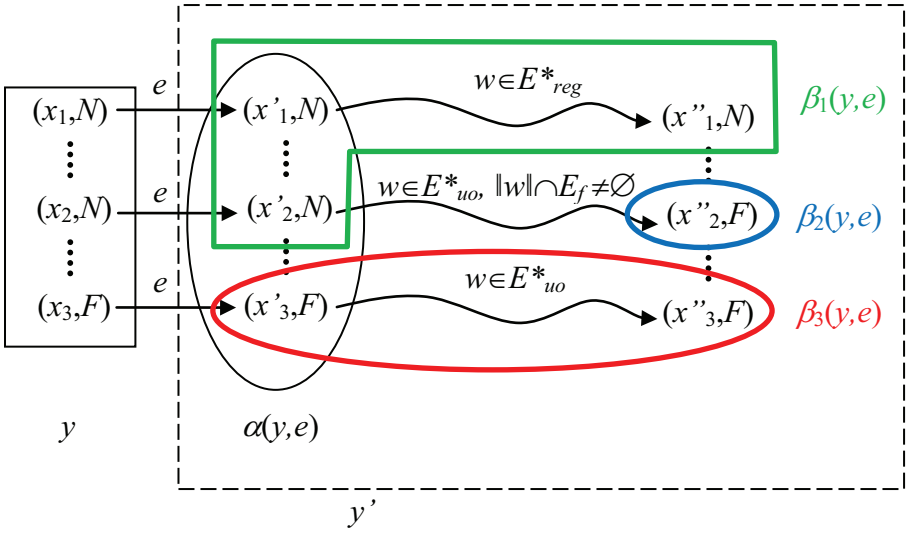

Figure 4: The computation of a new state $y' = \delta_y(y, e)$ of the diagnoser in step 4.(a) of Algoritm 1.

- Set $\alpha(y, e)$ contains the pairs (x', γ) such that, with $(x, \gamma) \in y$ and $e \in E$, state x' is reachable in G from state x executing exactly one e-transition.
- Set $\beta_1(y, e)$ contains the pairs (x'', N) such that, with $(x', N) \in \alpha(y, e)$, state x'' is reachable in G from state x' executing a sequence of regular transitions. In fact, if x' is reachable without a fault, such is also x''.
- Set $\beta_2(y, e)$ contains the pairs (x'', F) such that, with $(x', N) \in \alpha(y, e)$, state x'' is reachable in G from state x' executing a sequence of unobservable transitions that contains a fault. In this case, even if x' is reachable without a fault, state x'' can be reached with a fault.
- Set $\beta_3(y, e)$ contains the pairs (x'', F) such that, with $(x', F) \in \alpha(y, e)$, state x'' is reachable in G from state x' executing a sequence of unobservable transitions. In this case, since x' is reachable with a fault, such is also x''.

ii. **Let** $y' = \beta_1(y, e) \cup \beta_2(y, e) \cup \beta_3(y, e)$ and define $\delta_Y(y, e) = y'$. i.e., the occurrence of event e from state y of $Diag(G)$ yields y'.

iii. **If** $y' \notin Y \cup Y_{new}$ **then** $Y_{new} = Y_{new} \cup \{y'\}$.

(b) **Let** $Y = Y \cup \{y\}$ and $Y_{new} = Y_{new} \setminus \{y\}$.

5. **If** $Y_{new} \neq \emptyset$ **then goto** 4. ■

In Figure 4 a graphical description of the sets computed in step 4.(a) of the algorithm is shown.

We conclude with the following remark.

Proposition 1 *Given a plant G with state set X of cardinality n_x, let $Diag(G)$ be its diagnoser with state set Y of cardinality n_y. It holds $n_y < 2^{2n_x}$.*

Proof. Each state in Y is a non empty subset of elements in the set $Z = (X \times \{N\}) \cup (X \times \{F\})$ of cardinality $2n_x$. The number of possible subsets of Z including the empty set is 2^{2n_x}. □

3 Diagnosability

Let us now define a fundamental property relative to fault diagnosis.

Definition 6 A DFA G with alphabet $E = E_o \cup E_{uo}$ and set of fault event $E_f \subseteq E_{uo}$ is *diagnosable* if for all strings $ue_f \in L(G)$ such that $e_f \in E_f$ there exists a non negative integer $n \in \mathbb{N}$ such that

$$s = ue_f v \in L(G),\ |v| \geq n \quad \Longrightarrow \quad \nexists s' \in L(G) \cap (E \setminus E_f)^* \text{ such that } P(s) = P(s').$$

▲

This property can also be expressed as follows. Assume that the plant can generate a string ue_f that contains a fault and the evolution continues. After a finite number of steps n (that may depend on ue_f), when the new observed word is $s = ue_f v$ there exists no other string s' in the language of the plant that contains no fault and generates the same observation of s. This ensures that whenever a fault event e_f occurs, after a finite number of steps we will detect its occurrence because we will observe a word that is not consistent with any fault free string.

Problem 2 *Given a DFA G with alphabet $E = E_o \cup E_{uo}$ and set of fault event $E_f \subseteq E_{uo}$, the* diagnosability problem *consists in determining if G is diagnosable.*

We will show that the diagnoser, that provides a solution to the diagnosis problem, can also be a useful tool to solve the diagnosability problem. First, however, we need to introduce some definitions.

Definition 7 Given a diagnoser $Diag(G)$, a cycle

$$y_{j_1} \xrightarrow{e_1} y_{j_2} \xrightarrow{e_2} y_{j_3} \cdots y_{j_k} \xrightarrow{e_k} y_{j_1}$$

is called an *uncertain cycle* if all its states are uncertain, i.e., $\varphi(y_{j_i}) = U$ for $i = 1, 2, \ldots, k$. ▲

As a preliminary result, we can now state a sufficient condition for diagnosability.

Proposition 2 *A DFA G is diagnosable if its diagnoser does not contain uncertain cycles.*

Proof. Assume the DFA is not diagnosable. Then the following situation must occur:

- the DFA can generate a string $s = u\varepsilon_f$ containing fault ε_f;
- string s can be extended for an arbitrary length generating words $s_k = u\varepsilon_f e_1 e_2 \ldots e_k$ for $k \geq 1$
- there exists a fault free string $s'_k \in (E \setminus E_f)^*$ such that $P(s_k) = P(s'_k)$ for $k \geq 1$.

This means that in the diagnoser the observed word $w = P(s)$ yields an uncertain state y_{j_1} and from that state, as k grows, there exists words $w_k = P(s_k)$ of unbounded length (by Assumption A2) that will always yield an uncertain state. Since the number of states of the diagnoser is finite, this is only possible if there exists a cycle of uncertain states. □

Example 6 Consider again the DFA in Figure 1 whose diagnoser was shown in Figure 3. One can see that there exist in this diagnoser the 6 elementary cycles shown below (we have also reported the diagnosis state of each state along the cycle for a better understanding):

$$y_1\,[U] \xrightarrow{c} y_3\,[N] \xrightarrow{a} y_1\,[U] \qquad y_3\,[N] \xrightarrow{b} y_3\,[N] \qquad y_2\,[F] \xrightarrow{b} y_4\,[F] \xrightarrow{a} y_2\,[F]$$

$$y_2\,[F] \xrightarrow{c} y_4\,[F] \xrightarrow{a} y_2\,[F] \qquad y_2\,[F] \xrightarrow{a} y_2\,[F] \qquad y_4\,[F] \xrightarrow{b} y_4\,[F]$$

None of these cycles is uncertain, hence we conclude that the DFA is diagnosable. ◇

Next example shows that this sufficient condition for diagnosability is not necessary however.

Example 7 Consider the DFA in Figure 5 (left) where the set of observable events is $E_o = \{a, b\}$, the set of regular events is empty and the set of fault events is $E_f = \{\varepsilon_f\}$. The diagnoser for this DFA is shown in Figure 5 (right). One can see that there exists in the diagnoser a cycle of uncertain states

$$y_1\,[U] \xrightarrow{b} y_2\,[U] \xrightarrow{a} y_1\,[U]$$

However one can easily verify that this DFA is diagnosable. To show this, let us observe that two different type of faulty sequences may occur.

- Faulty sequences starting with $(ab)^k\varepsilon_f$. In this case, after the fault the system reaches state x_2 and in just two steps, when the sequence aa occurs, the observed word is $s = (ab)^k aa$. Since $P^{-1}(s) = \{(ab)^k\varepsilon_f aa\}$ there exists no fault free string s' consistent with this observation and that fault occurrence is detected.

- Faulty sequences starting with $(ab)^k a\varepsilon_f$. In this case, after the fault the system reaches state x_3 and in just two steps, when the sequence bb occurs, the observed word is $s = (ab)^k abb$. Since $P^{-1}(s) = \{(ab)^k a\varepsilon_f bb\}$ there exists no fault free string s' consistent with this observation and that fault occurrence is detected.

Hence the presence of an uncertain cycle in the diagnoser does not necessarily mean that we can have an observation of unbounded length *after the fault* that is consistent with both fault free and faulty strings. ◇

To derive a necessary and sufficient condition for diagnosability we introduce an additional concept.

Definition 8 (Refined sequence associated to an uncertain cycle) Given a diagnoser $Diag(G)$, consider an uncertain cycle

$$uc = y_{j_1} \xrightarrow{e_1} y_{j_2} \xrightarrow{e_2} y_{j_3} \xrightarrow{e_3} \quad \cdots \quad \xrightarrow{e_{k-1}} y_{j_k} \xrightarrow{e_k} y_{j_1}.$$

Let $y^1_{j_1}$ be the *refined diagnoser state* obtained from y_{j_1} removing all non faulty pairs (x, N). A *refined sequence* associated to uc is a sequence of diagnoser states obtained by applying the diagnoser construction from $y^1_{j_1}$ for repeated occurrences of the string of events $e_1 e_2 \cdots e_k$:

$$y^1_{j_1} \xrightarrow{e_1} y^1_{j_2} \xrightarrow{e_2} y^1_{j_3} \xrightarrow{e_3} \quad \cdots \quad \xrightarrow{e_{k-1}} y^1_{j_k} \xrightarrow{e_k} y^2_{j_1} \xrightarrow{e_1} y^2_{j_2} \xrightarrow{e_2} \cdots$$

It is not difficult to show that a refined sequence of diagnoser states as defined above:

- **either** will reach a state $y^k_j = y^{k+1}_j$ and hence can be continued indefinitely;

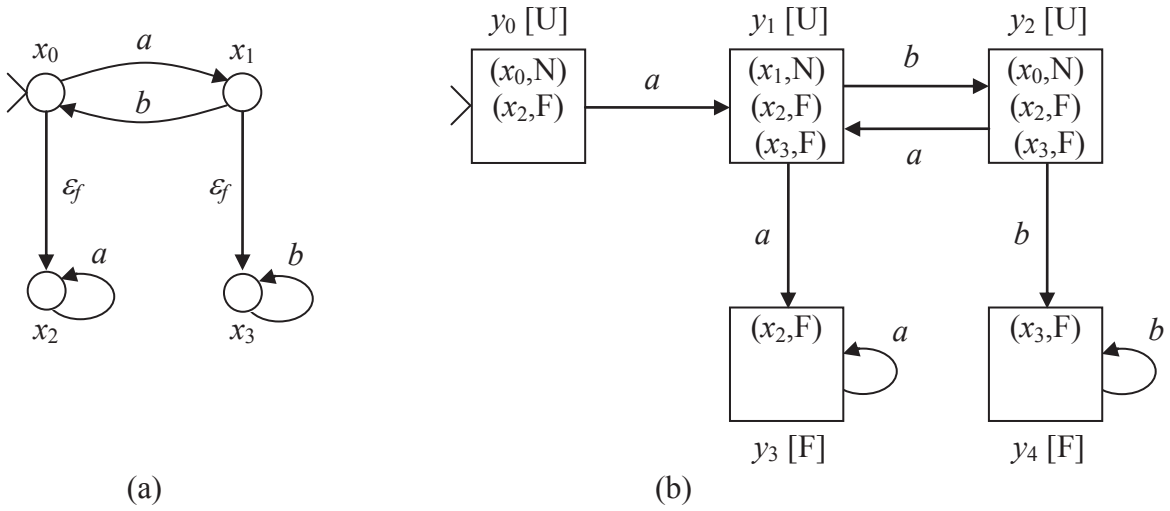

Figure 5: The DFA in Example 7 (a) and its diagnoser (b).

- **or** will eventually halt, reaching a state without a successor.

Definition 9 (Undeterminate cycle) An uncertain cycle uc is called an *undeterminate cycle* if its refined sequence can be continued indefinitely. ▲

We can finally present the following result whose proof follows from [2].

Proposition 3 *A DFA G is diagnosable if and only if its diagnoser $Diag(G)$ does non contain undeterminate cycles.* ■

Example 8 Consider again the DFA Figure 5 and studied in Example 7. We have already pointed out that there exists in the diagnoser a single uncertain cycle shown in Figure 6(a):

$$y_1\ [U] \xrightarrow{b} y_2\ [U] \xrightarrow{a} y_1\ [U]$$

where $y_1 = \{(x_1, N), (x_2, F), (x_3, F)\}$ and the cyclic sequence is ba

To construct the refined sequence of diagnoser states, we start from the refined diagnoser state $y_1^1 = \{(x_2, F), (x_3, F)\}$ obtained from y_1 removing the pair (x_1, N). We proceed to construct the refined sequence by repeated occurrences of sequence ba.

After the occurrence of event b we reach state $y_2^1 = \{(x_3, F)\}$ (see Figure 6(b)) from which event a cannot occur and the refined sequence halts. Hence the unique uncertain cycle of the diagnoser is not undeterminate. We conclude that the system is diagnosable, as already discussed in Example 7. ◇

Finally we present an example of a non diagnosable DFA.

Example 9 Consider the DFA in Figure 7(a) where the set of observable events is $E_o = \{a, b\}$, the set of regular events is $E_{reg} = \{\varepsilon_1\}$ and the set of fault events is $E_f = \{\varepsilon_f\}$. The diagnoser for this DFA is shown in Figure 7(b). One can see that there exists in the diagnoser a single uncertain cycle shown in Figure 8(a)

$$y_0\ [U] \xrightarrow{a} y_1\ [U] \xrightarrow{b} y_0\ [U]$$

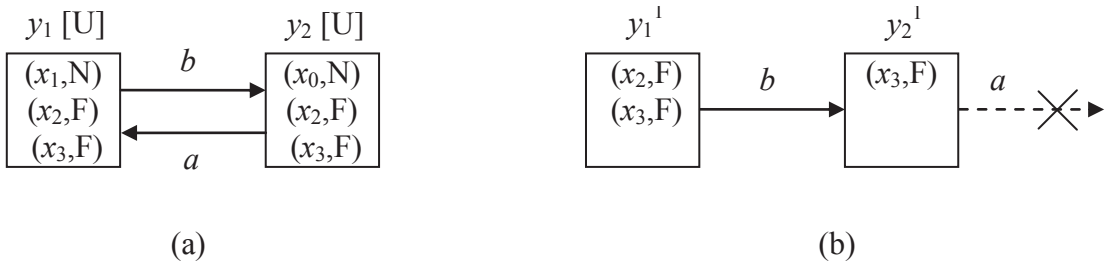

Figure 6: Uncertain cycle (a) and refined sequence (b) in Example 8.

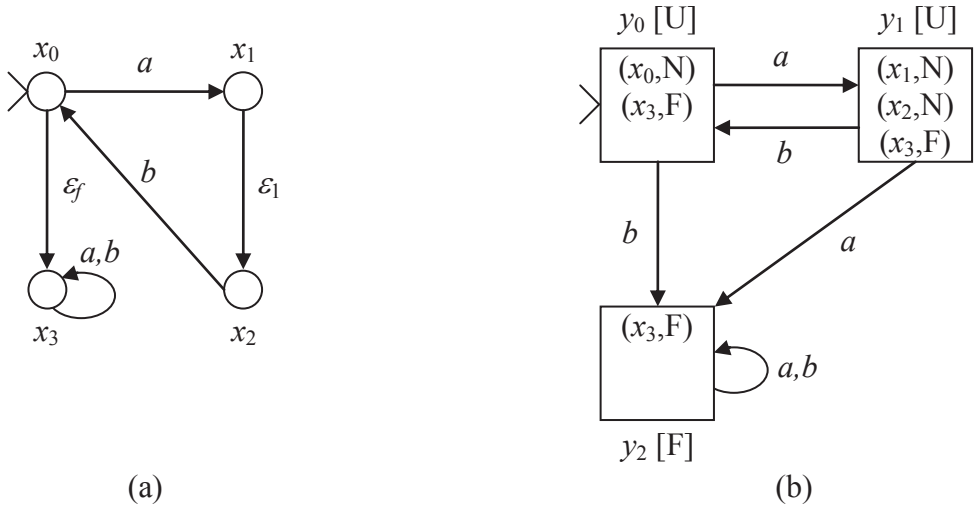

Figure 7: The DFA in Example 9 (a) and its diagnoser (b).

where $y_0 = \{(x_0, N), (x_3, F)\}$ and the cyclic sequence is ab.

To construct the refined sequence of diagnoser states, we start from the refined state $y_0^1 = \{(x_3, F)\}$ obtained from y_0 removing the pair (x_0, N). We proceed to construct the refined sequence by repeated occurrences of sequence ab.

After the occurrence of event a we reach state $y_1^1 = \{(x_3, F)\}$ from which the occurrence of event b yields $y_0^2 = \{(x_3, F)\}$ (see Figure 8(b)). Since $y_0^1 = y_0^2$ the refined sequence can be continued indefinitely. Hence the unique uncertain cycle of the diagnoser is undeterminate. We conclude that the system is not diagnosable.

Note in fact that the two strings $\varepsilon_f(ab)^k$ and $(a\varepsilon_1 b)^k$ produce the same observation. This means that if string $\varepsilon_f(ab)^k$ is generated by the system, after the fault we will have an observation of unbounded length that always produces an uncertain diagnosis state and does not allow to detect the occurrence of the fault. $\diamond$

References

[1] Cassandras, C.G., S. Lafortune, *Introduction to discrete event systems*, Springer, 2008.

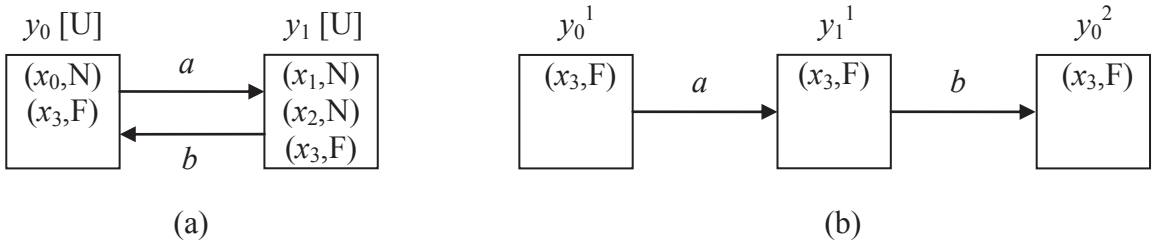

Figure 8: Uncertain cycle (a) and refined sequence (b) in Example 9.

[2] Sampath, M., R. Sengupta, S. Lafortune, K. Sinnamohideen, D. Teneketzis, "Diagnosability of Discrete-Event Systems," *IEEE Transaction on Automatic Control*, vol. 40, n. 9, pp. 1555–1575, September 1995.