Computation of minimal event bases that ensure diagnosability

João Carlos Basilio, Saulo Telles Souza Lima, Stéphane Lafortune & Marcos Vicente Moreira

Discrete Event Dynamic Systems
Theory and Applications

ISSN 0924-6703
Volume 22
Number 3

Discrete Event Dyn Syst (2012)
22:249-292
DOI 10.1007/s10626-012-0129-z

Springer

Discrete Event Dyn Syst (2012) 22:249–292
DOI 10.1007/s10626-012-0129-z

Computation of minimal event bases that ensure diagnosability

João Carlos Basilio · Saulo Telles Souza Lima · Stéphane Lafortune · Marcos Vicente Moreira

Received: 4 April 2011 / Accepted: 6 January 2012 / Published online: 29 January 2012

Abstract We deal with the problem of finding sets of observable events (event bases) that ensure language diagnosability of discrete-event systems modeled by finite state automata. We propose a methodology to obtain such event bases by exploiting the structure of the diagnoser automaton, and in particular of its indeterminate cycles. We use partial diagnosers, test diagnosers, and other new constructs to develop rules that guide the update of the observable event set towards achieving diagnosability. The contribution of this paper is the description of such rules and their integration into a set of algorithms that output minimal diagnosis bases.

Keywords Discrete event systems · Fault diagnosis · Sensor selection

1 Introduction

We study the sensor selection problem for ensuring the property of diagnosability for discrete-event systems modeled by finite-state automata. The property of diagnosability refers to the ability to detect the occurrence of unobservable events, such as faults, on the basis of observed traces of events and using model-based

J. C. Basilio (✉) · S. T. S. Lima · M. V. Moreira
COPPE, Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, 21949-900, Rio de Janeiro, R.J, Brazil
e-mail: basilio@dee.ufrj.br

S. T. S. Lima
e-mail: saulotelles@poli.ufrj.br

M. V. Moreira
e-mail: moreira@dee.ufrj.br

S. Lafortune
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA
e-mail: stephane@eecs.umich.edu

inferencing. Specifically, an unobservable event is diagnosable if every occurrence of it can be detected, after a bounded number of events, by a diagnostic engine driven by the observed events of the automaton; this property must hold over the entire language generated by the automaton. The study of formal diagnosability properties for discrete-event systems originated in the mid-1990s (see in particular Lin 1994; Sampath et al. 1995). Since then, a large amount of literature has been published on both theory and applications of diagnosability analysis; for a small sample of this work, the reader is referred to Pandalai and Holloway (2000), Sampath (2001), Sengupta (2001), Sinnamohideen (2001), Tripakis (2002), Boel and van Schuppen (2002), Jiang and Kumar (2004), Lunze and Schroder (2004), Garcia and Yoo (2005), Pencolé and Cordier (2005), Thorsley and Teneketzis (2005), Fabre et al. (2005), Wang et al. (2007), Genc (2008), Jéron et al. (2008), Kumar and Takai (2009), Cabasino et al. (2010), Haar (2010), and to the references contained therein.

The definition of diagnosability considered in this paper is the same as in Sampath et al. (1995). Our focus is on the design of the set of observable events in order to ensure that diagnosability holds. This falls in the category of sensor selection problems. In contrast to recent work on dynamic sensor activation in diagnosis problems (see, e.g., Thorsley and Teneketzis 2007, Cassez and Tripakis 2008, Wang et al. 2010 and Dallal and Lafortune 2010), we consider the "static" sensor selection problem, where the observability properties of an event are fixed over all system trajectories. This problem has been considered in the past literature, primarily from a computational viewpoint. In its simplest form, one needs to construct a set of observable events of minimal cardinality such that diagnosability holds. A brute-force approach to solving this problem involves testing the property of diagnosability over all subsets of the set of potentially observable events, denoted by Σ_o. It is assumed that the system is diagnosable with Σ_o. While the property of diagnosability can be tested in polynomial time in the size of the automaton modeling the system (see Jiang et al. 2001, Yoo and Lafortune 2002 and Moreira et al. 2011), the number of subsets to consider grows exponentially with the cardinality of Σ_o. In fact, it was shown in Yoo and Lafortune (2002) that the corresponding decision problem ("Does there exist a set of less than or equal to K observable events such that the system is diagnosable?") is NP-complete. To mitigate the computational efforts, various approaches have been proposed that exploit a *monotonicity* property of diagnosability in static sensor selection problems: If a system is diagnosable with observable event set A, then it will also be diagnosable with observable event set $B \supset A$; conversely, if it is not diagnosable with observable event set A, then it will not be diagnosable with observable event set $B \subset A$. This property implies the existence of *minimal* event sets that ensure diagnosability: A is such a minimal if the system is diagnosable under A but not diagnosable under any $B \subset A$. The monotonicity property is exploited in Jiang et al. (2003) to obtain a linear-time algorithm in the cardinality of Σ_o that results in a minimal observable event set (although not necessarily of minimum cardinality), and in Debouk et al. (2002) in the context of a stochastic version of the optimal sensor selection problem.

Our focus in this paper is on constructing minimal sets of observable events that ensure diagnosability; we call such sets *minimal diagnosis bases*. We propose a methodology for this construction that exploits structural properties of the system, as captured in the transition structure of diagnoser automata. Diagnoser automata, or simply diagnosers, are deterministic automata whose states are subsets of labeled

system states and whose events are the observable events of the system. The state label captures the occurrence or non-occurrence, in reaching the state, of the events to-be-diagnosed. For simplicity, and without loss of generality, we assume there is a single event to diagnose, denoted by σ_f; in this case, the label can be either N (for "no") or Y (for "yes"). Diagnosers were first proposed in Sampath et al. (1995) for testing the property of diagnosability. This test involves the detection of cycles that satisfy certain specific properties; these cycles are called indeterminate cycles. By examining the structure of the diagnoser, and in particular of its indeterminate cycles, it is possible to discover rules that can guide the update of the observable event set towards achieving diagnosability. The contribution of this paper is the discovery of such rules and their integration into a set of algorithms that output minimal diagnosis bases. To the best of our knowledge, these rules and associated algorithms are the first of their kind in the study of diagnosability of discrete event systems modeled by automata.

We note that the algorithms proposed in this paper output diagnosis bases and their corresponding diagnosers as well. These diagnosers can then be deployed for on-line diagnosis, as needed. If a brute force approach were employed to discover (minimal) diagnosis bases, the search would be exponential in the cardinality of Σ_o, as mentioned above, and worst-case polynomial in the state space of the system (if using verifiers) or worst-case exponential in the state space of the system (if using diagnosers); in the former case, the construction of the diagnosers for the identified diagnosis bases would be worst-case exponential in the state space of the system. Since our algorithms employ diagnosers, they are also exponential in the state space of the system, in the worst case. Our search over the subsets of Σ_o is however guided by structural properties, as captured in the constructed diagnoser automata. Instead of exhaustively testing all subsets of Σ_o, we will test potentially much fewer subsets, but at the price of additional calculations for identifying "promising" subsets. These additional calculations will be described in our technical development in Section 5.

Two types of diagnosers are defined and employed in this paper: partial diagnosers and test diagnosers. Partial diagnosers are constructed using a proper subset of Σ_o as set of observable events, while test diagnosers are obtained by parallel composition of partial diagnosers with the diagnoser corresponding to Σ_o. Partial unfoldings of these diagnosers are then built as trees and certain relevant paths in these trees, called faulty paths and prime paths, are characterized. Candidates for diagnosis bases are inferred from the tree built from the diagnoser for Σ_o; we call such sets *elementary diagnosing event sets*. The prime paths associated with the unfoldings are used to identify candidate events for growing the elementary diagnosis event sets until a minimal diagnosis basis is achieved. Our algorithms can be used to construct all minimal diagnosis bases, if so desired. Unlike enumerative approaches that search over all subsets of Σ_o, our approach exploits the transition structure of the system.

This paper is organized as follows. Section 2 presents necessary background on diagnosability. Section 3 discusses partial diagnosers and the notion of hidden indeterminate cycles; a preliminary version of the results in this section appears in Section III of Basilio and Lafortune (2009). Section 4 introduces the test diagnoser and presents an algorithm for the computation of elementary diagnosing event sets. The main algorithms for the construction of minimal diagnosis bases are developed in Section 5. Rules guiding the selection of events to include as observable events are demonstrated, based on tree unfoldings of the partial and test diagnosers and

associated prime paths. A brief conclusion follows in Section 6. We present two tables in the Appendix: the first one (Table 1) lists all the acronyms and the second one (Table 2) presents the main notation used in the paper.

2 Theoretical background

2.1 Definitions and notation

Let

$$G = (X, \Sigma, f, \Gamma, x_0) \tag{1}$$

denote a deterministic automaton, where X is the state space, Σ is the set of events, $f : X \times \Sigma \to X$ is the partial transition function, $\Gamma : X \to 2^{\Sigma}$ is the active event function, x_0 is the initial state of the system. In addition, assume that the set of events Σ is partitioned into two subsets: Σ_o, the set of observable events, *i.e.*, the set of events whose occurrence can be observed, and Σ_{uo}, the set of unobservable events. The unobservable events of the system are those events whose occurrence cannot be recorded by sensors, and also the failure events. Therefore, model G accounts for the normal and failure behaviors of the system.

Definition 1

A. The post-language of L after s is denoted by L/s, and is defined as

$$L/s = \{t \in \Sigma^* : st \in L\}. \tag{2}$$

B. The language projection P_o is defined in the usual manner (Ramadge and Wonham 1989), as

$$\begin{aligned} P_o : \Sigma^* &\to \Sigma_o^* \\ s &\mapsto P_o(s), \end{aligned} \tag{3}$$

with the following properties:

$$\begin{aligned} P_o(\epsilon) &= \epsilon, \\ P_o(\sigma) &= \begin{cases} \sigma, & \text{if } \sigma \in \Sigma_o \\ \epsilon, & \text{if } \sigma \in \Sigma_{uo} \end{cases}, \\ P_o(s\sigma) &= P(s)P(\sigma), s \in \Sigma^*, \sigma \in \Sigma, \end{aligned} \tag{4}$$

where ϵ denotes the empty trace. The inverse projection operator P_o^{-1} is defined as

$$P_o^{-1}(t) = \{s \in \Sigma^* : P_o(s) = t\}. \tag{5}$$

Both the projection and the inverse projection operations can be extended to languages in a straightforward way by applying $P_o(s)$ and $P_o^{-1}(s)$ to all $s \in L$.

C. Let $\Psi(\Sigma_f)$ denote the set of all traces in L that end with the failure event σ_f. Formally,

$$\Psi(\Sigma_f) = \{s \in L : s_f \in \Sigma_f\}, \tag{6}$$

where s_f denotes the last event of s. With slight abuse of notation, given a trace s, the membership relation $\Sigma_f \in s$ is used to denote that $\bar{s} \cap \Psi(\Sigma_f) \neq \emptyset$, where $\bar{s}$ denote the prefix-closure of s.

D. (Faulty trace) A trace $s \in L$ is a faulty trace if $\Sigma_f \in s$.

E. (Path and cyclic path) A path in G is a sequence $(x_1,\sigma_1,x_2,\ldots,\sigma_{n-1},x_n)$, where $\sigma_i \in \Sigma$, $x_{i+1} = f(x_i,\sigma_i)$, $i = 1,2,\ldots,n-1$. The path is cyclic if $x_1 = x_n$.

F. (Cycle) States $x_1,x_2,\ldots,x_n$ of X forms a cycle in an automaton G if there exists a trace $s = \sigma_1\sigma_2\ldots\sigma_n$ that originates in state x_1 such that $f(x_l,\sigma_l) = x_{l+1}$, $l = 1,\ldots,n-1$, and $f(x_n,\sigma_n) = x_1$.

G. (Unobservable reach) The unobservable reach of a state $x \in X$ with respect to a set Σ_{uo}, denoted by $UR(x,\Sigma_{uo})$, is defined as

$$UR(x,\Sigma_{uo}) = \{y \in X : (\exists t \in \Sigma_{uo}^*)[f(x,t) = y]\}.$$

This definition is extended to sets of states $B \subseteq X$ as follows:

$$UR(B,\Sigma_{uo}) = \bigcup_{x\in B} UR(x,\Sigma_{uo}).$$

□

Let us now define the following operation involving sets.

Definition 2 (Union product) The union product of sets Σ_i, $i = 1,2,\ldots,n$, denoted as $\Sigma_1 \dot{\times} \Sigma_2 \dot{\times} \ldots \dot{\times} \Sigma_n$, is defined as follows:

$$\Sigma_1 \dot{\times} \Sigma_2 \dot{\times} \ldots \dot{\times} \Sigma_n = \begin{cases} \{\Sigma_e = \Sigma_{e,1} \cup \Sigma_{e,2} \cup \ldots \cup \Sigma_{e,n} : \Sigma_{e,i} \in \Sigma_i, i = 1,2,\ldots,n\}, \text{ if} \\ \text{the elements of } \Sigma_i \text{ are sets,} \\ 2_1^{\Sigma_1} \dot{\times} 2_1^{\Sigma_2} \dot{\times} \ldots \dot{\times} 2_1^{\Sigma_n}, \text{ otherwise,} \end{cases}$$

$2_1^{\Sigma} = \{\tilde{\Sigma} \in 2^{\Sigma} : |\tilde{\Sigma}| = 1\}$, with $|.|$ denoting cardinality.

To illustrate the operations presented in Definition 2, let $\Sigma_1 = \{a,b\}$, $\Sigma_2 = \{b,c\}$, $\Sigma_3 = \{b\}$, and $\Sigma_4 = \{a,c\}$, and define $\Sigma_a = \{\Sigma_1, \Sigma_2\}$, $\Sigma_b = \{\Sigma_3, \Sigma_4\}$, and $\Sigma_c = \{\Sigma_4\}$. Then

$$\Sigma_a \dot{\times} \Sigma_b \dot{\times} \Sigma_c = \{\Sigma_1 \cup \Sigma_4, \Sigma_2 \cup \Sigma_4, \Sigma_1 \cup \Sigma_3 \cup \Sigma_4, \Sigma_2 \cup \Sigma_3 \cup \Sigma_4\} = \{\{a,b,c\}\}.$$

Consider now the product $\Sigma_1 \dot{\times} \Sigma_2 \dot{\times} \Sigma_3 \dot{\times} \Sigma_4$. Then

$$\Sigma_1 \dot{\times} \Sigma_2 \dot{\times} \Sigma_3 \dot{\times} \Sigma_4 = 2_1^{\Sigma_1} \dot{\times} 2_1^{\Sigma_2} \dot{\times} 2_1^{\Sigma_3} \dot{\times} 2_1^{\Sigma_4} = \{\{a,b\}, \{b,c\}, \{a,b,c\}\},$$

since $2_1^{\Sigma_1} = \{\{a\}, \{b\}\}$, $2_1^{\Sigma_2} = \{\{b\}, \{c\}\}$, $2_1^{\Sigma_3} = \{\{b\}\}$, and $2_1^{\Sigma_4} = \{\{a\}, \{c\}\}$.

2.2 Fault diagnosis of discrete event systems

Roughly speaking, the language generated by an automaton is diagnosable with respect to a set of observable events and a failure set $\Sigma_f \subseteq \Sigma_{uo}$ if the occurrence of any failure in Σ_f can be detected, within a finite delay, using only traces of observable events. The set of failure events Σ_f is usually partitioned into different subsets Σ_{f_i}, $i = 1,2,\ldots,m$, not necessarily singleton sets, so that each set Σ_{f_i} accounts for specific fault types; the reader is referred to Sampath et al. (1995, 1996) and Lafortune et al.

(2001) for more insight into this subject. Let $\Pi_f = \{\Sigma_{f_1}, \Sigma_{f_2}, \ldots, \Sigma_{f_m}\}$ denote this partition. Then, every time we say that a failure of type F_i has occurred, it is to be understood that some event from the set Σ_{f_i} has occurred.

In the study of fault diagnosis of DES, the following assumptions are usually made:

A1. The language generated by G is live, *i.e.*, $\Gamma(x_i) \neq \emptyset$ for all $x_i \in X$;
A2. Automaton G has no cyclic paths formed with unobservable events only.
A3. There is only one failure event, *i.e.*, $\Sigma_f = \{\sigma_f\}$.

Assumptions A1 and A3 are made for the sake of simplicity. Assumption A2 will be removed later with the introduction of the so-called hidden cycles.

Formally, diagnosability is defined as follows (Sampath et al. 1995).

Definition 3 A prefix-closed and live language L, generated by an automaton G, is diagnosable with respect to projection P_o and $\Sigma_f = \{\sigma_f\}$ if the following holds true:

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

where the diagnosability condition D is

$$(\forall \omega \in P_o^{-1}(P_o(st)) \cap L)(\Sigma_f \in \omega),$$

with $\|.\|$ denoting the length of a trace.

It is clear from Definition 3 that all traces of L that contain the faulty trace event σ_f must not have the same projection as any normal trace of L. This leads to the definition of ambiguous trace, as follows.

Definition 4 (Ambiguous trace) A faulty trace $s \in L$ is an ambiguous trace with respect to projection P'_o and σ_f if there exists a trace $\omega \in L$ such that $\Sigma_f \notin \omega$ and $P'_o(s) = P'_o(\omega)$.

It is important to remark that whereas the faulty trace s must have unbounded length, the length of the normal trace ω that makes s an ambiguous trace can be bounded.

One way to perform diagnosability verification is by using a deterministic automaton called diagnoser. The diagnoser of G, here denoted as G_d, has as events, the observable events of G and its states have labels Y and N attached to the states of G to indicate whether event σ_f has occurred or not. Formally, the diagnoser automaton G_d is defined as

$$G_d = (X_d, \Sigma_o, f_d, \Gamma_d, x_{0_d}), \tag{7}$$

and it can be computed in two steps: (i) perform the parallel composition $G \| A_\ell$, where A_ℓ is the two state label automaton shown in Fig. 1, and $\|$ denotes parallel composition; (ii) compute $Obs(G \| A_\ell, \Sigma_o)$, the observer of $G \| A_\ell$ with respect to Σ_o,

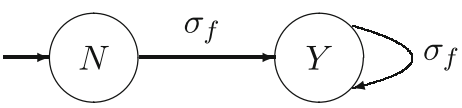


Fig. 1 Fault label automaton A_ℓ

i.e., assuming Σ_o as the set of observable events (Cassandras and Lafortune 2008). It is important to remark that the automaton obtained after the parallel composition performed in (i) generates the same language as G and also that its states are of the form (x,Y) or (x,N), depending on whether or not σ_f is in the traces that take x_0 to x; therefore $X_d \subseteq 2^{X\times\{N,Y\}}$.

When a diagnoser reaches a state whose labels are all Y, then it is certain that a fault has occurred, and if it is in a state whose labels are all N, it is certain that the system is in a normal path, *i.e.*, there is no fault occurrence. It is also possible for a diagnoser to be in a state that has both Y and N labels; in this case the diagnoser is uncertain about the fault occurrence. It is easy to see that, since $G_d = Obs(G\|A_\ell,\Sigma_o)$, then, once the diagnoser becomes certain about fault occurrence, it is not possible for it to become uncertain again; although it may be possible for a diagnoser to change from a non-faulty state to either an uncertain or certain (or faulty) state. This discussion leads to the following definitions for the diagnoser states, as far as the presence of labels Y and N are concerned (Sampath et al. 1995).

Definition 5 A state $x_d \in X_d$ is said to be certain (or faulty), if $\ell = Y$ for all $(x,\ell) \in x_d$, and normal (or non-faulty) if $\ell = N$ for all $(x,\ell) \in x_d$. If there exist $(x,\ell),(y,\tilde{\ell}) \in x_d$, x not necessarily distinct from y such that $\ell = Y$ and $\tilde{\ell} = N$, then x_d is an uncertain state of G_d.

Definition 6 A set of uncertain states $x_{d_1},x_{d_2},\ldots,x_{d_n} \in X_d$ forms an indeterminate cycle if the following conditions hold true:

(1) $x_{d_1},x_{d_2},\ldots,x_{d_n}$ form a cycle in G_d, *i.e.*, there exists $\sigma_l \in \Sigma_o$, $l = 1,2,\ldots,n$, such that $f_d(x_{d_l},\sigma_l) = x_{d_{l+1}}$, $l = 1,2,\ldots,n-1$, and $f_d(x_{d_n},\sigma_n) = x_{d_1}$;

(2) $\exists(x_l^{k_l},\ell_l^{k_l}),(\tilde{x}_l^{r_l},\tilde{\ell}_l^{r_l}) \in x_{d_l}$, $x_l^{k_l}$ not necessarily distinct from $\tilde{x}_l^{r_l}$, $l = 1,2,\ldots,n$, $k_l = 1,2,\ldots,m_l$, and $r_l = 1,2,\ldots,\tilde{m}_l$ such that

   (a) $\ell_l^{k_l} = Y$, $\tilde{\ell}_l^{r_l} = N$, for all l, k and r;

   (b) The sequences of states $\{x_l^{k_l}\}$, $l = 1,2,\ldots,n$, $k_l = 1,2,\ldots,m_l$ and $\{\tilde{x}_l^{r_l}\}$, $l = 1,2,\ldots,n$, $r_l = 1,2,\ldots,\tilde{m}_l$ can be rearranged to form cycles in G, such that the corresponding traces s and $\tilde{s}$, formed with the events that define the evolution of the cycles, have as projection $\sigma_1\sigma_2\ldots\sigma_n$, where σ_1, σ_2, and σ_n are defined in (1).

Using Definitions 3 and 6, the following necessary and sufficient condition for language diagnosability can be stated.

Theorem 1 (Sampath et al. 1995) *A language L generated by an automaton G is diagnosable with respect to projection P_o and $\Sigma_f = \{\sigma_f\}$ if, and only if, its diagnoser G_d has no indeterminate cycles.*

Remark 1 The diagnosability of the language generated by G is, according to Definition 3, based solely on a single set of observable events, or equivalently, on the projection $P_o : \Sigma^* \to \Sigma_o^*$. This means that, in practice, the decision on whether a fault has occurred or not is taken by one central diagnoser. For this reason, this problem is usually referred in the literature to as the centralized diagnosis problem and G_d is referred to as a central diagnoser.

3 Diagnosability under partial observation

Definition 3 of language diagnosability takes into account not only the language generated by an automaton but also the set of observable events and the failure partition. The dependence of language diagnosability on the set of observable events suggests that it may be possible that the language generated by an automaton be also diagnosable with respect to another projection $P'_o : \Sigma^* \to \Sigma_o'^*$, where $\Sigma'_o \subset \Sigma_o$ and Σ_f. This problem is referred to as centralized diagnosability under partial observation. In order to address this problem, we make another assumption.

A4. L is diagnosable with respect to projection $P_o : \Sigma^* \to \Sigma_o^*$ and Σ_f (centralized diagnosable).

Let $G'_d = (X'_d, \Sigma'_o, f'_d, \Gamma'_d, x'_{0_d})$ denote a diagnoser for L assuming partial observation, *i.e.*, G'_d is capable of observing only events in a set $\Sigma'_o \subset \Sigma_o$. Such a diagnoser will be referred throughout the text to as a central diagnoser with partial observation or simply partial diagnoser. The result that follows shows that if G_d has been computed, then it is not necessary to compute G'_d from G but directly from G_d.

Theorem 2 *The partial diagnoser G'_d and $\hat{G}'_d = Obs(G_d, \Sigma'_o) = (\hat{X}'_d, \Sigma'_o, \hat{f}'_d, \hat{\Gamma}'_d, \hat{x}'_{0_d})$ (the observer of G_d with respect to projection $P_{oo'} : \Sigma_o^* \to \Sigma_o'^*$) are equal up to the following renaming of states:*

$$\hat{x}'_d = \{x_{d_1}, x_{d_2}, \ldots, x_{d_n}\} \in \hat{X}'_d,\ x_{d_i} \in X_d \Leftrightarrow x'_d = \cup_{i=1}^n x_{d_i} \in X'_d. \tag{8}$$

Proof Let $\Sigma = \Sigma_o \cup \Sigma_{uo}$, and consider the non-empty set $\Sigma'_o \subset \Sigma_o$. Define:

(i) $G_\ell = G \| A_\ell = (X_\ell, \Sigma, f_\ell, \Gamma_\ell, x_{0_\ell})$;
(ii) $G_d = Obs(G_\ell, \Sigma_o) = (X_d, \Sigma_o, f_d, \Gamma_d, x_{0_d})$;
(iii) $G'_d = Obs(G_\ell, \Sigma'_o) = (X'_d, \Sigma'_o, f'_d, \Gamma'_d, x'_{0_d})$.

We need to prove that $\hat{G}'_d$ and G'_d are isomorphic.

Since $P_{oo'}[P_o(s)] = P'_o(s), \forall s \in L$, we may conclude that $L(\hat{G}'_d) = L(G'_d)$. Therefore, we only need to prove the state equivalence of Eq. 8. In order to do so, let us consider any $s' \in L(\hat{G}'_d) = L(G'_d)$. For that s' there exist two corresponding states $\hat{x}'_d = \hat{f}'_d(\hat{x}'_{0_d}, s')$ and $x'_d = f'_d(x'_{0_d}, s')$.

Let us consider, initially, state $\hat{x}'_d$. Then, for each $x_{d_i} \in \hat{x}'_d$, $i \in \{1,2,\ldots,n\}$, there exists a trace $s_{d_i} \in L(G_d)$ such that $P_{oo'}(s_{d_i}) = s'$ and $f_d(x_{0_d}, s_{d_i}) = x_{d_i}$. Analogously, for each $x_\ell \in x_{d_i}$, there exists a trace $s \in L(G_\ell)$ such that $P_o(s) = s_{d_i}$ and $f_\ell(x_{0_\ell}, s) = x_\ell$. Therefore, since G'_d is a deterministic automaton and $P_{oo'}[P_o(s)] = P'_o(s)$, there exists a state $x'_d = f'_d(x'_{0_d}, s')$ such that $x_\ell \in x'_d$. This implies that, for all $x_\ell \in x_{d_i} \in \hat{x}'_d$, $x_\ell \in x'_d$, and, thus, $\cup_{i=1}^n x_{d_i} \subseteq x'_d$.

Let us now consider state x'_d. Then, there exist a trace $s \in L(G_\ell)$ such that $s' = P'_o(s)$, and a state $x_\ell = f_\ell(x_{0_\ell}, s)$ such that $x_\ell \in x'_d$. Since $\hat{G}'_d$ is a deterministic automaton and $P_{oo'}[P_o(s)] = s'$, there exist states $x_{d_i} = f'_d(x'_{0_d}, P_o(s))$ and $\hat{x}' = \hat{f}'_d(\hat{x}'_{0_d}, s')$ such that $x_\ell \in x_{d_i} \in \hat{x}$. Therefore, $x'_d \subseteq \cup_{i=1}^n x_{d_i}$. □

According to Theorem 2, the partial diagnoser G'_d that observes the events in a subset Σ'_o of the set of observable events Σ_o can be built from the full observation

diagnoser G_d simply by merging the states of G_d that are connected by the events in $\Sigma_o \setminus \Sigma'_o$ into a single state formed by the union of the sets of the states merged. As a consequence, even though, due to Assumption A2, the languages generated by centralized diagnosers with full observation are always live, the language generated by a partial diagnoser is not necessarily live. This happens whenever the events of a cyclic path in G_d become unobservable in the partial diagnoser; it is not difficult to see that when this happens, this cycle reduces to a single state in G'_d.

When unobservable events occur after the system reaches a state that has been obtained by merging states that form cycles, then, although there is no change of states in the partial diagnoser following the occurrence of unobservable events, the actual states of the automaton changes cyclically. In this case, it is said that such a partial diagnoser has a hidden cycle, whose formal definition is as follows.

Definition 7 (Hidden cycles and indeterminate hidden cycles) Let $x'_d \in X'_d$ be obtained by merging states $x_{d_1}, x_{d_2}, \ldots, x_{d_n} \in X_d$. Then there exists a hidden cycle in x'_d in G'_d if, for some $\{i_1, i_2, \ldots, i_k\} \subset \{1,2,\ldots,n\}$, $x_{d_{i_1}}, x_{d_{i_2}}, \ldots, x_{d_{i_k}}$ form a cycle in G_d. Moreover, if x'_d is uncertain and all states $x_{d_{i_1}}, x_{d_{i_2}}, \ldots, x_{d_{i_k}}$ are certain, then the hidden cycle is indeterminate. □

Remark 2

(a) The introduction of hidden cycles allows us to remove Assumption A2 from this point onwards.
(b) Due to Assumption A4, L is diagnosable with respect to P_o and Σ_f which implies that G_d has no indeterminate cycles; the only cycles that may appear in G_d are formed with certain, normal or uncertain states that do not form indeterminate cycles. Therefore, states $x_{d_{i_k}}$, $k = 1,2,\ldots,n$, that form a hidden cycle in x'_d, must all be certain, normal or uncertain.
(c) Hidden cycles will be represented in the state transition diagrams of partial diagnosers by dashed self-loops: indeterminate hidden cycles will be labeled as *ihc* and all other hidden cycles will be labeled simply as *hc*, since, as it will be seen in the sequel, they do not interfere in the diagnosability under partial observation.
(d) From this point onwards, in order to differentiate between indeterminate cycles that are not hidden and those that can be observed in a diagnoser, the latter will be referred to as indeterminate observed cycles.

The following Theorem provides a necessary and sufficient condition for diagnosability under partial observation.

Theorem 3 *Assuming that a language L is diagnosable with respect to projection P_o and Σ_f, then L will be also diagnosable with respect to projection P'_o, $\Sigma'_o \subset \Sigma_o$, and $\Sigma_f = \{\sigma_f\}$ if, and only if, G'_d has no indeterminate cycles (observed or hidden).*

Proof A necessary and sufficient condition for diagnosability when G'_d has no hidden cycles can be established by following the same strategy as in the proof of Theorem 1. Let us now consider the case when G'_d has indeterminate hidden cycles.

Let $x_{d_{YN}}$ be an uncertain state of G_d and define $\hat{x}'_d = UR(x_{d_{YN}}, \Sigma_o \setminus \Sigma'_o)$. Form x'_d by renaming $\hat{x}'_d$ according to Eq. 8. Then, there always exists an uncertain state $x'_{d_{YN}}$

of G'_d such that $x'_{d_{YN}} \subseteq x'_d$. Now, assume that, for $l = 1, \ldots, n$, states $x_{d_{Y,l}} \in x'_{d_{YN}}$ form an indeterminate hidden cycle in $x'_{d_{YN}}$. It is not hard to see that there exists a trace $w_k = stu_k \in L$ that satisfies the following conditions

(1) $s \in \Psi(\Sigma_f)$ and $f_d(x_{0_d}, P_o(s)) = x_{d_{YN}}$;
(2) $t \in (\Sigma_o \setminus \Sigma'_o)^*$ is such that $f_d(x_{0_d}, P_o(st)) = x_{d_{Y,1}}$;
(3) $u_k \in (\Sigma_o \setminus \Sigma'_o)^*$, $\|u_k\| = k$, with k arbitrarily large, is such that $f_d(x_{d_{Y,1}}, u_k) = x_{d_{Y,(k \bmod n)+1}}$.

Consequently, $\Sigma_f \in w_k$, $f'_d(x_{0_d}, P'_o(stu_k)) = f'_d(x_{0_d}, P'_o(s)) = x'_{d_{YN}}$, and $P'_o(s) = P'_o(w_k)$. Finally, since $x_{d_{YN}}$ is an uncertain state, there exists $w \in L$ such that $\Sigma_f \notin w$ and $P'_o(w) = P'_o(w_k)$, which violates the diagnosability condition.

For the reverse direction, it is clear from the proof of Theorem 1, given in Sampath et al. (1995), that if G'_d has no indeterminate (observed or hidden) cycles then the language is diagnosable. This is so because when L is not diagnosable, the two traces that cause the violation of diagnosability will lead to indeterminate cycles in G'_d that are either observed or hidden. □

Example 1 Consider automaton G depicted in Fig. 2, where $\Sigma = \{a,b,c,d,\sigma,\sigma_f\}$, $\Sigma_o = \{a,b,c,d\}$, $\Sigma_{uo} = \{\sigma,\sigma_f\}$ and $\Sigma_f = \{\sigma_f\}$ is the failure event set. The diagnoser G_d

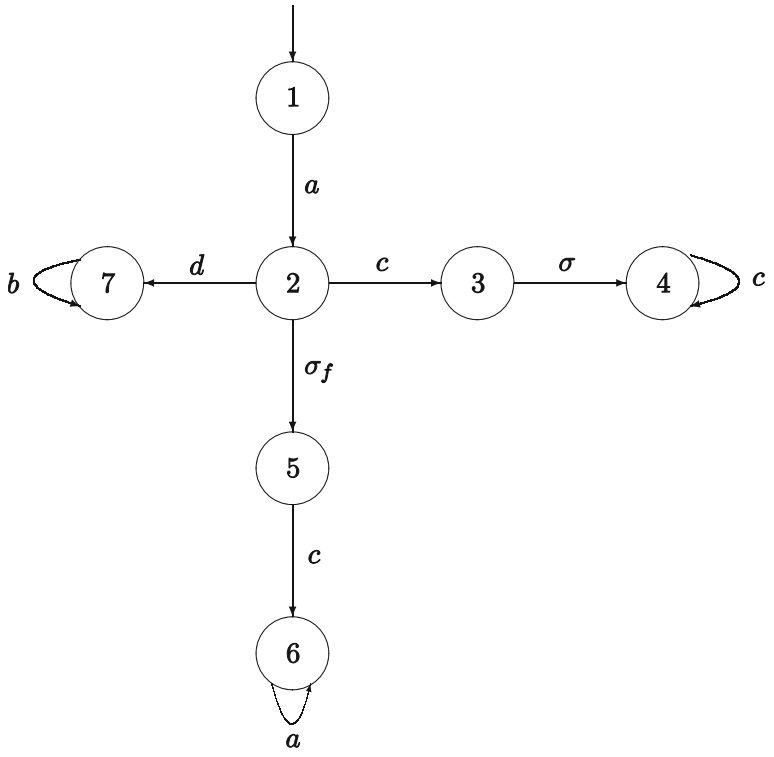


Fig. 2 Automaton G for Example 1

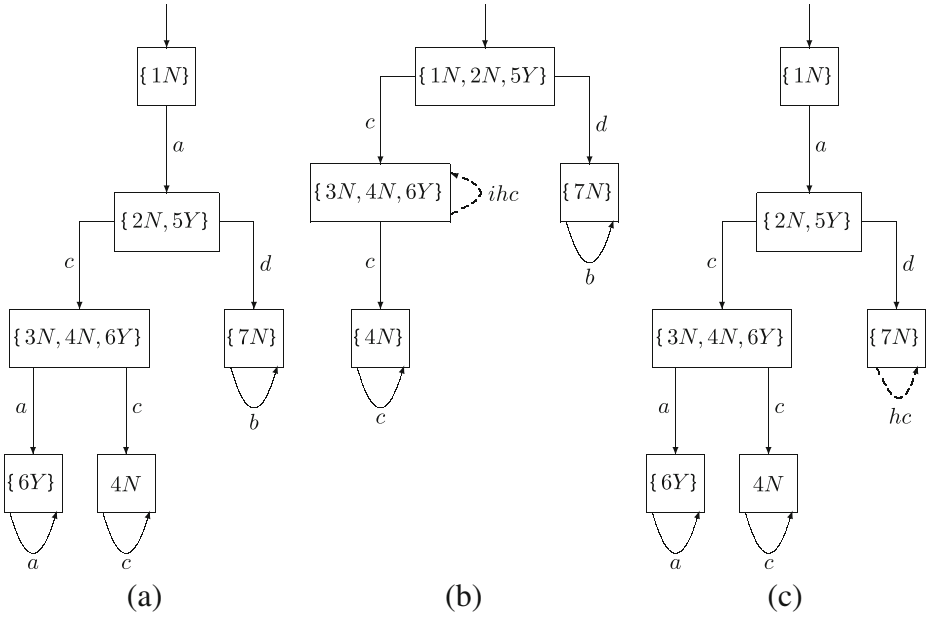


Fig. 3 Diagnoser G_d (**a**) and partial diagnosers G'_d (**b**) and G''_d (**c**) for the following sets of observable events: $\Sigma'_o = \{c,d\}$ and $\Sigma''_o = \{a,c,d\}$

for G is shown in Fig. 3a. Since G_d has no indeterminate cycles, it can be concluded that L is diagnosable with respect to P_o and Σ_f.

Consider now the problem of checking whether L is also diagnosable with respect to projection P'_o and Σ_f, where $\Sigma'_o = \{c,d\} \subset \Sigma_o$. The partial diagnoser G'_d, assuming Σ'_o as the set of observable events, is shown in Fig. 3b. Note that since G'_d has an indeterminate hidden cycle in state $\{3N,4N,6Y\}$, L is not diagnosable with respect to P'_o and Σ_f. The reason for the non-diagnosability of L with respect to P'_o is the existence of a faulty trace $s = a\sigma_f ca^n$, $n \in \mathbb{N}$, that has the same projection over P'_o as the normal trace $s' = ac$, *i.e.*, $P'_o(s) = P'_o(s') = c$; therefore s is an ambiguous trace (s is actually the unique ambiguous trace in this example). Note that since event $a \in s$ but $a \notin \Sigma'_o$, then, by adding event a to the set of observable events Σ'_o, and forming a new set of observable events $\Sigma''_o = \{a,c,d\}$, we may expect that L becomes diagnosable with respect to $P''_o : \Sigma^* \to \Sigma''^*_o$ and Σ_f. This is actually true, as seen in Fig. 3b, since G''_d has no indeterminate cycles (observed or hidden).

4 Diagnosis bases for diagnosability

4.1 Elementary diagnosing event sets

The results presented in Theorem 3 lead directly to the following questions. Do there exist different subsets of the set of observable events for which the language generated by an automaton is diagnosable? What is the minimum cardinality subset

of the set of observable events capable of diagnosing the language generated by an automaton? The answer to these questions starts with the following definitions.

Definition 8 (Diagnosis basis) A set $\Sigma'_o \subset \Sigma_o$ is a diagnosis basis for L if L is diagnosable with respect to projection P'_o and $\Sigma_f = \{\sigma_f\}$.

Definition 9 (Minimal diagnosis basis) A set $\Sigma'_o \subset \Sigma_o$ is a minimal diagnosis basis for L if Σ'_o is a diagnosis basis but, for any non-empty proper subset Σ''_o of Σ'_o, L is not diagnosable with respect to projection P''_o and $\Sigma_f = \{\sigma_f\}$.

According to Definitions 8 and 9, the main difference between diagnosis and minimal diagnosis bases is with respect to the nature of the events. The events in a minimal diagnosis basis are all essential, in the sense that the language generated is no longer diagnosable when any event is removed from the basis. On the other hand, a non-minimal diagnosis basis has redundant events, in the sense that, not all events in the basis set are necessary to diagnose the fault occurrence.

Using Definitions 8 and 9, the problem of finding all sets $\Sigma'_o \subset \Sigma_o$ for which L is diagnosable with respect to P'_o can also be stated as follows: given an automaton $G = (X, \Sigma, f, \Gamma, x_0)$, where $\Sigma = \Sigma_o \cup \Sigma_{uo}$, and assuming that Σ_o is a diagnosis basis, find all sets $\Sigma'_o \in 2^{\Sigma_o} \setminus \{\Sigma_o, \emptyset\}$ that are also diagnosis bases.

One possible way to solve this problem is by using a brute force method, which consists of forming set $\mathcal{P}(\Sigma_o) = 2^{\Sigma_o} \setminus \{\emptyset, \Sigma_o\}$ and to test for each set $\Sigma'_o \in \mathcal{P}(\Sigma_o)$ if L is diagnosable with respect to Σ'_o and Σ_f. In order to find minimal diagnosis bases in a brute force manner, one would start by examining subsets of observable events of cardinality one, then move on to subsets of cardinality two, as so forth. We propose, instead, to exploit the structure of the system, as captured in its diagnoser, in order to more efficiently search for minimal diagnosis basis candidates over $\mathcal{P}(\Sigma_o)$. Specifically, our idea for choosing initial candidates for minimal diagnosis bases is as follows. Since L is, by assumption, diagnosable with respect to P_o and Σ_f, the diagnoser G_d has no indeterminate cycles, which implies that there must exist at least one subtrace of events that takes uncertain states of G_d to some cycle of certain states. Therefore, at least one event of each one of these subtraces must be observable in order for the language to be diagnosable; otherwise there would exist an indeterminate hidden cycle in the uncertain state making the language non-diagnosable. This idea is formally developed below.

Let $x_{d_{YN}}, x_{d_Y}, x_{d_N} \in X_d$ denote, respectively, uncertain, certain and normal states of G_d. Due to Assumption A4, it is always possible to define the following subset of X_d:

$$X^Y_{YN} = \{x_{d_{YN}} \in X_d : (\exists (x_{d_Y}, \sigma) \in X_d \times \Sigma_o)[\, f_d(x_{d_{YN}}, \sigma) = x_{d_Y}]\}. \tag{9}$$

Note that since L is, by assumption, live, for each state of X^Y_{YN}, it is always possible to form at least one path $P_Y = (x_{d_{YN}}, \sigma_0, x_{d_{Y,1}}, \sigma_1, \ldots, \sigma_{n-1}, x_{d_{Y,n}})$ satisfying the following conditions: (i) $x_{d_{Y,n}} = x_{d_{Y,i}}$ for some $i \in \{1, 2, \ldots, n-1\}$, that is, $(x_{d_{Y,i}}, \sigma_i, x_{d_{Y,i+1}}, \ldots, \sigma_{n-1},\ x_{d_{Y,n}})$ form a cyclic path; (ii) $(x_{d_{Y,i}}, \sigma_i, x_{d_{Y,i+1}}, \ldots, \sigma_{n-1}, x_{d_{Y,n}})$ is the only cyclic path in P_Y. As a consequence, the set X^Y_{YN} will be referred here to as a faulty path origin state set (FPOSS) and path P_Y as a faulty path. The elements of X^Y_{YN} are called faulty path origin states (or simply origin states, where there is no danger of misunderstanding).

Definition 10 (Faulty path event, faulty path event set)

A. An event $\sigma \in \Sigma_o$ is a faulty path event if it belongs to any faulty path defined for any state of X_{YN}^{Y}.
B. A faulty path event set (FPES), denoted as Σ_{fpes}, is a set formed with all events of a faulty path.

The definition of faulty path event sets allows us to derive a necessary condition for a set $\Sigma_o' \subset \Sigma_o$ to be a diagnosis basis, as follows.

Proposition 1 *Let N_{fpes} denote the number of faulty path event sets of G_d. Then a necessary condition for $\Sigma_o' \subset \Sigma_o$ to be a diagnosis basis for L and $\Sigma_f = \{\sigma_f\}$ is that*

$$\Sigma_o' \cap \Sigma_{\text{fpes},i} \neq \emptyset, i = 1,2,\ldots,N_{\text{fpes}}. \tag{10}$$

Proof Let Σ_o' be a diagnosis basis and assume that for some $k \in \{1,2,\ldots,N_{\text{fpes}}\}$, $\Sigma_{\text{fpes,k}} \cap \Sigma_o' = \emptyset$. Therefore, for some $x_{d_{YN}} \in X_{YN}^{Y}$, there exists a faulty path $P_Y^k = (x_{d_{YN}}, \sigma_0^k, \ x_{d_{Y,1}}^k, \sigma_1^k, \ldots, \sigma_{n-1}^k, x_{d_{Y,n}}^k)$, satisfying $x_{d_{Y,j}}^k = x_{d_{Y,n}}^k$ for some $j \in \{1,2,\ldots,n-1\}$. It is immediate to see that $x_{d_{Y,j}}^k, x_{d_{Y,j+1}}^k, \ldots, x_{d_{Y,n}}^k$ form an indeterminate hidden cycle in a state $x_{d_{YN}}' \in X_d'$ that contains $UR(x_{d_{YN}}, \Sigma_o \setminus \Sigma_o')$. According to Theorem 3, this implies that L is not diagnosable with respect to projection P_o' and $\Sigma_f = \{\sigma_f\}$, which contradicts the assumption that Σ_o' is a diagnosis basis. □

Remark 3 Note that the condition imposed by Proposition 1 is only necessary. As will be clarified in the examples to be presented later, it may be possible that condition 10 be satisfied, but Σ_o' is not a diagnosis basis. The necessary and sufficient condition for Σ_o' to be a diagnosis basis is that given in Theorem 3.

It is clear that in order for a fault occurrence to be diagnosed, at least one event in each faulty path must be observable. This leads to the definition of elementary diagnosing event sets (EDES).

Definition 11 (Elementary diagnosing event sets) Let $\Sigma_{\text{fpes},i}$, $i = 1,\ldots,N_{\text{fpes}}$ denote the faulty path event sets of G_d. The set of all elementary diagnosing event sets of G_d is defined as follows:

$$\Sigma_{\text{edes}} = \Sigma_{\text{fpes},1} \dot{\times} \Sigma_{\text{fpes},2} \dot{\times} \ldots \dot{\times} \Sigma_{\text{fpes},N_{\text{fpes}}}, \tag{11}$$

where the union product above is performed according to Definition 2.

Algorithm 1 provides a systematic way to find all elementary diagnosing event sets of G_d.

In step 1 of Algorithm 1, all faulty path origin states of G_d are identified and the set X_{YN}^{N} is computed. The trees formed in steps 2 to 4 arrive at a leaf whenever it reaches any first revisited state; therefore defining a faulty path since it initiates at a faulty path origin state of G_d and has a unique cyclic path. Therefore, all FPESs can be obtained directly from the edges of the branches of the trees. In step 5 all EDESs are formed by applying the union product to guarantee that each set has at least one event of each FPESs. However, since theses sets are to be used in the search

Algorithm 1 (Algorithm for finding all EDESs of G_d)

STEP 1 Build the centralized diagnoser G_d and find the FPOSS (X^Y_{YN}) of G_d. Let $|X^Y_{YN}| = N_{YN}$.

STEP 2 For each origin state $x_{d_{YN,i}} \in X^Y_{YN}, i = 1,2,\ldots,N_{YN}$ form a rooted tree[1] with root $x_{d_{YN,i}}$, as follows:

(i) Let $\Gamma^Y_d(x_{d_{YN,i}}) = \{\sigma \in \Gamma_d(x_{d_{YN,i}}) : f_d(x_{d_{YN,i}},\sigma) = x_{d_Y}\}$ and assume that $|\Gamma^Y_d(x_{d_{YN,i}})| = n_{YN,i}$. Create $n_{YN,i}$ proper descendants of $x_{d_{YN,i}}$ and label them as x_{d_Y}, where $x_{d_Y} = f_d(x_{d_{YN,i}},\sigma)$, $\sigma \in \Gamma^Y_d(x_{YN,i})$. Label the edge $(x_{d_{YN,i}},x_{d_Y})$ as σ;

(ii) A node labeled as x_{d_Y}, defined in the tree, will be a leaf if state x_{d_Y} has already labeled any proper ancestor of x_{d_Y}. Otherwise, let $|\Gamma_d(x_{d_Y})| = n_Y$. Create n_Y proper descendants of x_{d_Y} and label them as $x_{d_{Y,\text{new}}}$, where $x_{d_{Y,\text{new}}} = f_d(x_{d_Y},\sigma), \sigma \in \Gamma_d(x_{d_Y})$. Label the edge $(x_{d_Y},x_{d_{Y,\text{new}}})$ as σ.

STEP 3 For each tree T_i, $i = 1,2,\ldots,N_{YN}$, identify its leaves $x^\ell_{d_{Y,i}}$, $\ell = 1,\ldots,\ell_{T_i}$, where ℓ_{T_i} is the number of leaves in tree T_i. Form paths $P^\ell_{Y,i}$, $\ell = 1,\ldots,\ell_{T_i}$, starting at $x_{d_{YN,i}}$ and ending at $x^\ell_{d_{Y,i}}$, $\ell = 1,\ldots,\ell_{T_i}$ (these paths are actually the faulty paths starting at $x_{d_{YN,i}}$).

STEP 4 Form FPESs $\Sigma^\ell_{\text{fpes},i}$, $i = 1,\ldots,N_{YN}$, $\ell = 1,\ldots,\ell_{T_i}$ with each path $P^\ell_{Y,i}$ obtained in the previous step.

STEP 5 With the FPESs obtained in step 4, form the set of elementary diagnosing event sets according to Eq. 11.

STEP 6 Remove from Σ_{edes}, all event sets $\Sigma' \in \Sigma_{\text{edes}}$ for which there exists another set $\Sigma'' \in \Sigma_{\text{edes}}$ such that $\Sigma' \supseteq \Sigma''$.

of minimal diagnosis bases, those EDESs that are supersets of another EDES must be removed from the set. This is done in step 6.

Remark 4 (Computational complexity of Algorithm 1)
The computational complexity of Algorithm 1 will be discussed later in the paper (see Remarks 5 and 10).

The following example illustrates the computation of all elementary diagnosing event sets of a given centralized diagnoser.

Example 2 Consider automaton G depicted in Fig. 4a and assume that $\Sigma_o = \{a,b,c,d,e\}$ and $\Sigma_f = \{\sigma_f\}$. It is clear that the centralized diagnoser G_d, shown in Fig. 4b, has no indeterminate cycles and therefore L is diagnosable with respect to P_o and Σ_f.

[1] Strictly speaking, the graph to be built in Algorithm 1 is not a rooted tree since distinct nodes may have the same label. The main reason for labeling two distinct nodes with the same label is due to the fact that we are unfolding a directed graph (diagnoser), which has cycles and, in some cases, there is more than one path from an origin state to a certain state.

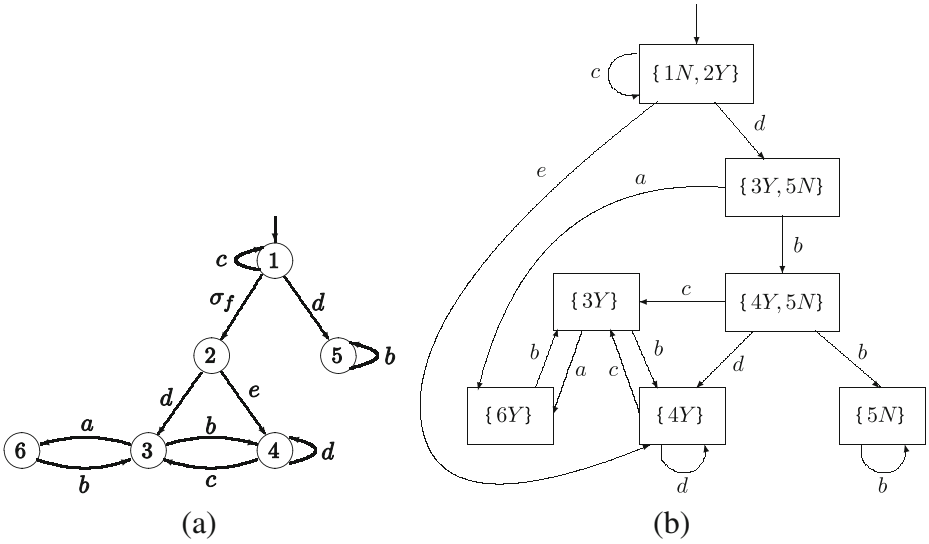


Fig. 4 Automaton and corresponding diagnoser for illustration of elementary diagnosing event sets

According to Algorithm 1, to find the elementary diagnosing event sets for L, the first step is to identify the origin states of G_d. It is clear from Fig. 4b that $X_{YN}^{Y} = \{x_{d_{YN,1}}, x_{d_{YN,2}}, x_{d_{YN,3}}\}$, where $x_{d_{YN,1}} = \{1N,2Y\}$, $x_{d_{YN,2}} = \{3Y,5N\}$ and $x_{d_{YN,3}} = \{4Y,5N\}$. The next step of Algorithm 1 is to build a tree for each of the origin states above, which are shown in Fig. 5. Based on the trees of Fig. 5, it is possible, as required in step 3 of Algorithm 1, to identify their leaves, and in the sequel to form the paths from the root to the leaves. In particular, the tree of Fig. 5c has six leaves, which allow us to obtain the following faulty paths: $P_{Y,3}^{1} = (\{4Y,5N\},c,\{3Y\},b,\{4Y\},c,\{3Y\})$, $P_{Y,3}^{2} = (\{4Y,5N\},c,\{3Y\},b,\{4Y\},d,\{4Y\})$, $P_{Y,3}^{3} = (\{4Y,5N\},c,\{3Y\},a,\{6Y\},b,\{3Y\})$, $P_{Y,3}^{4} = (\{4Y,5N\},d,\{4Y\},d,\{4Y\})$, $P_{Y,3}^{5} = (\{4Y,5N\},d,\{4Y\},c,\{3Y\},b,\{4Y\})$, and $P_{Y,3}^{6} = (\{4Y,5N\},d,\{4Y\},c,\{3Y\},a,\{6Y\},b,\{3Y\})$. Proceeding in the same way, six other paths can be obtained from the trees of Fig. 5a and b. Therefore, it is straightforward to see that the FPESs of G_d are given by: $\Sigma_{\text{fpes},1}^{1} = \{d,e\}$, $\Sigma_{\text{fpes},1}^{2} = \{b,c,e\}$, $\Sigma_{\text{fpes},1}^{3} = \{a,b,c,e\}$, $\Sigma_{\text{fpes},2}^{1} = \{a,b\}$, $\Sigma_{\text{fpes},2}^{2} = \{a,b,c\}$, $\Sigma_{\text{fpes},2}^{3} = \{a,b,d\}$, $\Sigma_{\text{fpes},3}^{1} = \{b,c\}$, $\Sigma_{\text{fpes},3}^{2} = \{b,c,d\}$, $\Sigma_{\text{fpes},3}^{3} = \{a,b,c\}$, $\Sigma_{\text{fpes},3}^{4} = \{d\}$, $\Sigma_{\text{fpes},3}^{5} = \{b,c,d\}$, and $\Sigma_{\text{fpes},3}^{6} = \{a,b,c,d\}$. Proceeding in accordance with step 5 of Algorithm 1, the following set of EDESs is obtained:

$$\Sigma_{\text{edes}} = \{\{a,b,d\}, \{a,b,c,d\}, \{b,d\}, \{b,c,d\}, \{a,b,d,e\}, \{a,b,c,d,e\}, \{b,d,e\}, \{b,c,d,e\}, \{a,c,d\}, \{a,c,d,e\}\}. \tag{12}$$

Since we are interested in the smallest cardinality sets, the set above can be reduced, in accordance with step 6, to:

$$\Sigma_{\text{edes}} = \{\{b,d\}, \{a,c,d\}\}. \tag{13}$$

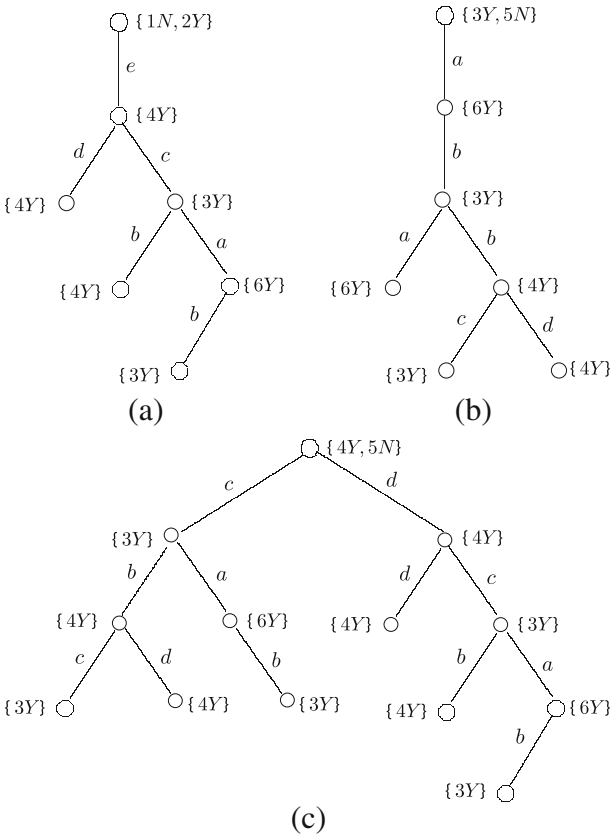


Fig. 5 Rooted trees with roots labeled as $x_{d_{YN,1}} = \{1N,2Y\}$ (**a**), $x_{d_{YN,2}} = \{3Y,5N\}$ (**b**), and $x_{d_{YN,3}} = \{4Y,5N\}$ (**c**)

4.2 A new diagnosability condition

According to Proposition 1, the elementary diagnosing event sets that have the smallest cardinality are the sets with the minimum number of events necessary to diagnose the occurrence of σ_f. It is also clear that the complete assessment of the diagnosability of L with respect to P'_o and $\Sigma_f = \{\sigma_f\}$, where $\Sigma'_o \in \Sigma_{\text{edes}}$, requires the construction of a partial diagnoser G'_d. According to Theorem 3, L is also diagnosable with respect to P'_o and $\Sigma_f = \{\sigma_f\}$ if, and only if, the partial diagnoser G'_d has no indeterminate cycles (observed or hidden). If for some Σ'_o, L is diagnosable then Σ'_o is a minimal diagnosis basis. However, when L is not diagnosable with respect to P'_o and $\Sigma_f = \{\sigma_f\}$, it is necessary to add new events to Σ'_o. Since we are looking for minimal diagnosis bases, the insertion of events must be done carefully in order to avoid adding redundant events. Section 5 will describe our approach for selecting the new events to be included. For that matter, in the remainder of this section, we introduce an automaton whose structure will be exploited so as to identify the events in $\Sigma_o \setminus \Sigma'_o$ that should be added to Σ_o to construct a minimal diagnosis basis.

Let us define automaton G'_{test} as follows:

$$G'_{\text{test}} = G'_d \| G_d = (X_t, \Sigma_o, f_t, \Gamma_t, x_{t_0}). \tag{14}$$

Note that the state x_t of G'_{test} has the following structure:

$$x_t = (x'_d, x_d),$$

where $x'_d \in X'_d$ and $x_d \in X_d$.

Definition 12 A state x_t of G'_{test} is certain if x'_d and x_d are both certain and uncertain if x_d is certain but x'_d is uncertain.

Definition 13 A cycle of uncertain states in G'_{test} is said to be indeterminate if the states of G'_d that appears in the first components of the states in the cycle also form an indeterminate cycle (observed or hidden) in G'_d.

Let L_d, L'_d and L'_{test} denote, respectively, the languages generated by G_d, G'_d and G'_{test}. From the definition of G'_{test} given in Eq. 14, it is clear that

$$L'_{\text{test}} = P^{-1}_{oo'}(L'_d) \cap L_d = L_d,$$

where $P^{-1}_{oo'}$ is with respect to Σ_o and not with Σ. It is well known that a necessary and sufficient condition for a language L not to be diagnosable with respect to projection P'_o and Σ_f is the existence of ambiguous traces t_Y with respect to P'_o and Σ_f. The following results show that G'_{test} can be used not only as a diagnosability test but also to find all ambiguous traces t_Y with respect to P'_o and Σ_f.

Theorem 4 *Assume that L is diagnosable with respect to projection P_o and $\Sigma_f = \{\sigma_f\}$. Then, L will be diagnosable with respect to the projection P'_o, $\Sigma'_o \subset \Sigma_o$, and $\Sigma_f = \{\sigma_f\}$ if, and only if, G'_{test} has no indeterminate cycles, where G'_{test} is defined according to Eq. 14.*

Proof

($\Rightarrow$) Assume that G'_{test} has an indeterminate cycle and consider a trace $st \in L$ that satisfies the following conditions: (i) $s \in \Psi(\Sigma_f)$; (ii) $\|t\| > n_t$, where n_t can be arbitrarily large; (iii) $P_o(st)$ cycles in an indeterminate cyclic path of G'_{test}. Let $s't' = P'_o(st)$. Then, due to the structure of G'_{test}, $s't'$ either cycles over an indeterminate cycle of G'_d or stops in an uncertain state of G'_d (when the indeterminate cycle of G'_{test} is associated with a hidden cycle in G'_d). This implies that $\exists w \in P'^{-1}_o[P'_o(st)] \cap L$ such that $\Sigma_f \notin w$, which violates the diagnosability Definition 5, or equivalently, that L is not diagnosable with respect to P'_o and $\Sigma_f = \{\sigma_f\}$.

($\Leftarrow$) Assume that G'_{test} has no indeterminate cycle and consider a trace $s \in \Psi(\Sigma_f)$. Since L is diagnosable with respect to P_o and Σ_f, then for all long enough traces t in L/s *i.e.*, $\|t\| > n_t$, where n_t can be arbitrarily large, $P_o(st)$ takes G'_{test} to a state $x_t = (x'_d, x_d)$ with x_d certain; the corresponding component x'_d of x_t may be either certain or uncertain. However, since G'_{test} has no indeterminate cycle, then t can be increased further so as to make x'_d also certain. This implies that $\exists s't' = P'_o(st)$ that leads to a certain state of G'_d. Since $s \in L$ is arbitrary, then L is also diagnosable with respect to P'_o and $\Sigma_f = \{\sigma_f\}$. □

Since L is diagnosable with respect to P_o and $\Sigma_f = \{\sigma_f\}$, the following result can be derived directly from Theorem 4.

Corollary 1 *Under the same assumptions as those of Theorem* 4*, an arbitrarily long trace* $s'_{\text{test}} \in L'_{\text{test}}$ *that loops in an indeterminate cycle of* G'_{test} *is such that* $P'^{-1}_o\{P'_o[P^{-1}_o(s'_{\text{test}}) \cap L]\} \cap L$ *has both faulty and non-faulty traces.* □

The implication of Corollary 1 is that even though s'_{test} is not, in general, a trace of L, since it is defined over Σ^*_o, it bears a close relationship with the ambiguous traces of L with respect to P'_o and Σ_f. In order to explain this fact, let L_d denote the language generated by G_d. According to Theorem 2, $P'_o(L) = P_{oo'}(L_d)$, and since L is diagnosable with respect to P_o and Σ_f, then the diagnosability analysis of L with respect to P'_o and Σ_f can be carried out by using L_d and $P_{oo'}(L_d)$ in place of L and $P'_o(L)$. In addition, the non-diagnosability of L with respect to P'_o and Σ_f is due to the existence of indeterminate cycles (hidden inclusive) in G'_d. A connection between these cycles of G'_d and their inverse projections in G'_{test} will be pursued in the sequel.

5 Searching for minimal diagnosis bases for diagnosability

Since the condition given by Proposition 1 is only necessary, it is very likely that a set $\Sigma'_o \in \Sigma_{\text{edes}}$ is not a diagnosis basis. Therefore, in order to obtain new minimal diagnosis basis candidates for L, it is necessary to add events to Σ'_o, *i.e.*, to find a set $\Sigma_{ies} \subseteq \Sigma_o \setminus \Sigma'_o$ and form a new set $\Sigma''_o = \Sigma'_o \cup \Sigma_{ies}$.

An immediate way to find event sets whose union with Σ'_o are minimal diagnosis bases is to carry out an exhaustive search over the set $2^{\Sigma_o \setminus \Sigma'_o} \setminus \{\emptyset, \Sigma_o \setminus \Sigma'_o\}$. However, this approach does not exploit the structural knowledge captured in G'_{test}, which provides for the identification of ambiguous traces of L with respect to $P_o{}'$. Instead, we form a new event set $\Sigma''_o = \Sigma'_o \cup \{\sigma\}$, where σ is either an event belonging to a trace $s_Y \in L_d$ ($s_Y = P_o(t_Y)$), where t_Y is an ambiguous trace of L with respect to projection P'_o, or an event belonging to a trace $s_N \in L_d$ ($s_N = P_o(t_N)$), where t_N is a normal trace of L, that satisfy $P_{oo'}(s_N) = P_{oo'}(s_Y)$; the latter will be the case when the chosen event either appears only in s_N or in certain positions of s_Y and s_N that make $P''_o(s_Y) \neq P''_o(s_N)$.

According to Theorems 3 and 4, the non-diagnosability of L with respect to P'_o and Σ_f implies that both G'_d and G'_{test} have one or more indeterminate cycles (the former may also have hidden cycles). Let us consider, initially, the indeterminate observed cycles of G'_d. It is not difficult to see that, in this case, there exist two arbitrarily long traces $t_Y, t_N \in L$ such that $\Sigma_f \in t_Y$ but $\Sigma_f \notin t_N$ whose corresponding projections $s_Y = P_o(t_Y)$ and $s_N = P_o(t_N)$ ($s_Y, s_N \in L_d$) satisfy the following conditions:

OC1. $f_d(x_{0_d}, s_Y) = x_{d_Y}$ and $f_d(x_{0_d}, s_N) = x_{d_N}$, where x_{d_Y} (x_{d_N}) is a certain (respectively, normal or uncertain) state of G_d that belongs to a cycle of certain (respectively, normal or uncertain states only), where, in the case of uncertain states, they do not form indeterminate cycles in G_d;

OC2. $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'_{YN}$, where s'_{YN} is such that $f'_d(x'_{0_d}, s'_{YN}) = x'_{d_{YN}}$, with $x'_{d_{YN}}$ belonging to an indeterminate cycle of G'_d.

Therefore, to each indeterminate observed cycle in G'_d, we may associate at least two cycles in G'_{test}, as follows: (i) a cycle formed with the states whose first components are the states $x'_{d_{YN}}$ of G'_d that are reached through s'_{YN} and whose second components are the certain states x_{d_Y} of G_d that are reached through s_Y; (ii) another cycle that is formed with states whose first components are the same states $x'_{d_{YN}}$ of the first cycle, and whose second components are either normal states or uncertain states of G_d that are not part of an indeterminate cycle that are reached through s_N.

Consider now the existence of indeterminate hidden cycles in G'_d. In this case, there always exist two traces $t_Y, t_N \in L$, where $\Sigma_f \in t_Y$ but $\Sigma_f \notin t_N$, whose corresponding projections $s_Y = P_o(t_Y)$ and $s_N = P_o(t_N)$ $(s_Y, s_N \in L_d)$ satisfying the following conditions:

HC1. s_Y arbitrarily long and s_N with bounded length;
HC2. $s' = P_{oo'}(s_Y) = P_{oo'}(s_N)$ is also bounded.

This is so because for a hidden cycle to be indeterminate, it must be formed with certain states that form a cycle in G_d, therefore guaranteeing the existence of a trace s_Y arbitrarily long. In addition, since indeterminate hidden cycles are formed in uncertain states, then it is always possible to find a bounded trace s_N that takes G_d from its initial state to either a normal or an uncertain state.

We may, therefore, conclude that, whether G'_d has an indeterminate observed cycle or an indeterminate hidden cycle, a necessary condition for L to be diagnosable with respect to P''_o and Σ_f, where $\Sigma''_o = \Sigma'_o \cup \Sigma_{ies}$ $(\Sigma_{ies} \subseteq \Sigma_o \setminus \Sigma'_o)$ is that Σ_{ies} possesses events of either s_Y or s_N that makes $P_{oo''}(s_Y) \neq P_{oo''}(s_N)$.

5.1 Prime paths, covering prime path and cover for a path with embedded cycles

Although the idea of adding to Σ'_o events belonging to $\Sigma_o \setminus \Sigma'_o$ that appear either in s_Y or s_N, where s_Y and s_N satisfy either conditions OC1. and OC2. or conditions HC1. and HC2., seems to be simple, this is usually a very difficult task since cycles are likely to have embedded cycles; for instance if states $x_{t_1}, x_{t_2}, x_{t_3}, x_{t_2}$ form a cycle, then it is possible to define several cycles of states with the states of this cycle (*e.g.* $(x_{t_1}, x_{t_2}, x_{t_3}, x_{t_2})$, (x_{t_1}, x_{t_2}) and $(x_{t_1}, x_{t_2}, x_{t_3}, x_{t_2}, x_{t_3}, x_{t_2})$). As a consequence, there may exist several traces s_Y and s_N, even when there exists a unique trace connecting the initial state of G'_{test} to the first state of the cycle. Therefore, the use of ambiguous traces to obtain events for Σ_{ies} requires that all traces that cycle in indeterminate cycles of G'_{test} be found. An immediate approach to this problem is to use the algorithm proposed in Johnson (1975), which finds all elementary circuits of directed graphs, to compute all elementary cyclic paths of the indeterminate cycles of G'_{test}. However, this approach would only be suitable for a special class of automata since Johnson's algorithm assumes that the directed graph does not have self-loops and multiple edges between the same vertices. Therefore, a different approach will be proposed here.

Let

$$P_l^c = (x_l, \sigma_l, x_{l+1}, \sigma_{l+1}, \ldots, \sigma_{n-1}, x_n, \sigma_n, x_l) \tag{15}$$

denote a path of an automaton G that has one or more embedded cyclic paths, namely that x_i is not necessarily different from x_j, $i \neq j$, $i, j \in \{l, l+1, \ldots, n\}$ and define a path

$$P_0 = (x_0, \sigma_0, x_1, \sigma_1, \ldots, x_{l-1}, \sigma_{l-1}, P_l^c), \tag{16}$$

where x_0 is the initial state of G. We start with the following definition.

Definition 14 (Prime path) Path P_0 defined according to Eqs. 16 and 15 is a prime path if $x_i \neq x_j$ for all $i \neq j$ and $i, j \in \{0,1,2,\ldots,n\}$.

The computation of all prime paths of an automaton G can be carried out through the construction of a rooted tree T with root x_0, similar to that obtained in accordance with Algorithm 1, as proposed in Algorithm 2.

Algorithm 2 (Algorithm for obtaining all prime paths of an automaton)

STEP 1 Label the root of T as x_0.

STEP 2 Let $|\Gamma(x_0)| = n_0$ and $x = f(x_0,\sigma)$, $\sigma \in \Gamma(x_0)$. Create n_0 proper descendants of x_0 and label them as x and the corresponding edge (x_0,x) as σ.

STEP 3 A node labeled as x, defined in the tree, will be a leaf if state x has already labeled any proper ancestor of x. Otherwise, let $|\Gamma(x)| = n$ and $x_{\text{new}} = f(x,\sigma)$, $\sigma \in \Gamma(x)$. Create n proper descendants of x and label them as x_{new} and the corresponding edge (x,x_{new}) as σ. Repeat this step until all states x_{new} give rise only to leaves.

STEP 4 Identify all leaves x_l of T and form all possible paths that start at the root and end at x_l.

Notice that any branch of the tree formed according to Algorithm 2 defines a path that starts at the initial state of the automaton and has a unique cyclic path, being therefore a prime path. Since all possible paths that can be followed from the initial state are considered, the algorithm returns all prime paths of G.

Remark 5 (Computational complexity of Algorithm 2)

Let E and N denote, respectively, the cardinality of the event set and the state space of automaton G whose prime paths we want to obtain using the tree described in Algorithm 2. Notice that all prime paths start at the initial state of G and so E^0 nodes are formed. Since there are E events in G and G is by assumption deterministic, at most E^1 nodes (states) can be obtained after the initial state. After that at most E nodes can be obtained for each node in the previous step, resulting in at most E^2 nodes. Since there exist N states, the maximum length of any prime path is N, which implies that the maximum number of nodes in the tree is $\sum_{k=0}^{N} E^k$. Therefore, the worst case complexity of finding all prime paths by building a tree as described in Algorithm 2 is of the order of E^N. This compares favorably with the upper bound on the number of elementary cycles in a complete directed graph given in Johnson (1975), which can be verified to be of the order of N^{N-1}. This is so because we are dealing with a deterministic automaton as opposed to a complete directed graph. In our case, E is likely to be much smaller than N; in fact, in discrete event modeling by parallel composition, E grows linearly in the number of system components while N grows exponentially in the number of system components. It is worth noting that the complexity of Algorithm 1 is essentially the same as that of Algorithm 2 since it is based on a similar tree construction where the initial state is replaced with the origin states and each branch stops at revisited states. In the sequel, Algorithm 2 will be employed in different contexts, where the automaton of interest will either be G'_d or G'_{test}; note that in both of these cases, E is upper bounded by $|\Sigma_o|$.

When P_0 has more than one embedded cycles, it is no longer a prime path. It is therefore necessary to split P_0 in an appropriate way so as to obtain prime paths that keep all information regarding the events that appear in all transitions of all embedded cycles. This leads to the definitions of covering prime paths and cover for a path with embedded cycles.

Definition 15 (Covering prime paths) Let P_0 be a path with embedded cycles as defined in Eqs. 16 and 15. A covering prime path of P_0 is a prime path obtained from P_0 by deleting some cycles in P_0.

Definition 16 (Cover for a path with embedded cycles) Let $C(P_0) = \{P_{0,1}, P_{0,2}, \ldots, P_{0,\eta}\}$ denote a set formed with η covering prime paths of P_0. Then $C(P_0)$ will be a cover for P_0 if and only if any transition defined in P_0 appears in at least one prime path of $C(P_0)$.

The definition of cover for a path with embedded cycles above generalizes that of cycle cover usually adopted in graph theory. An efficient algorithm for finding cycle covers has been proposed by Itai et al. (1981).

Remark 6 It is not difficult to see from Definitions 14, 15 and 16 that any prime path is a covering prime path and, as a consequence, a cover of itself.

Let us now apply the definitions introduced above to G'_{test}. Under the assumption that $\Sigma'_o \subset \Sigma_o$ is not a diagnosis basis, then G'_{test} always has indeterminate cycles, *i.e.*, cycles of states whose first components are uncertain states of G'_d and the second components are certain states of G_d. This leads us to the following definition.

Definition 17 (Y-prime paths of G'_{test}) A Y-prime path of G'_{test} is a prime path whose states of the unique cyclic path form an indeterminate cycle in G'_{test}.

The following result may be stated.

Proposition 2 *A path P_0 with embedded cyclic paths of G'_{test} has no embedded indeterminate cyclic paths if and only if it has no covering Y-prime paths.*

Proof

($\Rightarrow$) Let P_0 be a path with embedded cycles and assume that P_0 has no embedded indeterminate cyclic paths. Then, by slightly modifying Algorithm 2, it is not difficult to show that all prime paths that appear in this path can be obtained. Since, by assumption, none of the cyclic states correspond to an indeterminate cycle, all resulting prime paths have no indeterminate cycles and, thus, no Y-prime path is obtained.

($\Leftarrow$) Assume that P_0 has no covering Y-prime path but P_0 has embedded indeterminate cycles. Due to diagnoser construction, once G_d reaches a certain state, it is not possible for it to go back to a normal or an uncertain state, and therefore, when P_0 reaches a state belonging to an embedded indeterminate cyclic path it must cycle only over uncertain states of G'_{test}. Since all transitions must appear in all paths of any cover for P_0, then it is always possible to

obtain a covering Y-prime path in some cover $C(P_0)$, which contradicts the assumption that P_0 has no covering Y-prime path.

We may conclude, therefore, that, according to Proposition 2, the existence of indeterminate cycles in G'_{test} can be avoided if we guarantee that there is no prime path whose unique cyclic path is indeterminate. This is an important result since it replaces the search for paths with embedded indeterminate cyclic paths with the search for all prime paths whose unique cyclic path is indeterminate.

5.2 Dealing with indeterminate observed cycles

Let us consider initially the paths with embedded cycles of G'_{test} formed with states whose first components form indeterminate observed cycles in G'_d. As stated earlier, since L is diagnosable with respect to P_o and Σ_f, the presence of indeterminate observed cycles in G'_d is determined by the existence of, at least, two arbitrarily long traces $s_Y, s_N \in L_d$ satisfying conditions OC1. and OC2. In order to avoid that an indeterminate observed cycle of G'_d continues to exist in G''_d, where $\Sigma''_o = \Sigma'_o \cup \Sigma_{ies}$, $\Sigma_{ies} \subseteq \Sigma_o \setminus \Sigma'_o$, it is necessary and sufficient that $G''_{\text{test}} = G''_d \| G_d$ does not have any indeterminate cycle. Since, according to Proposition 2, any embedded cyclic path of G'_{test} has no embedded indeterminate cyclic paths if and only if it has no covering Y-prime paths, we must seek necessary conditions to prevent all Y-prime paths of G'_{test} from being Y-prime paths of G''_{test}.

Let us define the following sets:

$$S_Y = \{s : s \text{ is a trace associated with a Y-prime path of } G'_{\text{test}}\}, \tag{17}$$

$$S_N = \{s : s \text{ is a trace associated with a prime path of } G'_{\text{test}} \text{ whose first components of the states of the unique cyclic path are uncertain states of an indeterminate cycle (observed or hidden) of } G'_d \text{ and the second components are either normal states of } G_d \text{ or uncertain states of } G_d \text{ that are not states of an indeterminate cycle in } G'_d\}. \tag{18}$$

We state the following result.

Theorem 5 *Let us assume that L is not diagnosable with respect to P'_o and Σ_f and that G'_d has indeterminate observed cycles only. In addition, let s' denote a trace of L'_d formed with the events of a prime path whose unique cycle is indeterminate and observed. Then, it is always possible to find a pair of traces $(s_Y, s_N) \in S_Y \times S_N$ such that $s' \in \overline{P_{oo'}(s_Y)}$ and $s' \in \overline{P_{oo'}(s_N)}$.*

Proof Let s' be a trace of L'_d formed with the events of a prime path of G'_d whose unique cycle is indeterminate and observed and assume that it is not possible to find a pair of traces $(s_Y, s_N) \in S_Y \times S_N$ that satisfy $s' \in \overline{P_{oo'}(s_Y)}$ and $s' \in \overline{P_{oo'}(s_N)}$. In order for such a pair not to exist, one of the following conditions must hold true:

(i) $\forall s_N \in L'_{\text{test}} : s' \in \overline{P_{oo'}(s_N)}, \nexists s_Y \in L'_{\text{test}} : s' \in \overline{P_{oo'}(s_Y)}$;
(ii) $\forall s_Y \in L'_{\text{test}} : s' \in \overline{P_{oo'}(s_Y)}, \nexists s_N \in L'_{\text{test}} : s' \in \overline{P_{oo'}(s_N)}$.

Let us suppose, initially, that condition (i) above holds true. It is not hard to see that since $G'_{\text{test}} = G'_d \| G_d$ and $L'_{\text{test}} = L_d$, a state in a path of G'_{test} will be revisited through s only if both a state of G'_d is revisited through $P_{oo'}(s)$ and a state of G_d is revisited through s. As a consequence, for any trace $s_Y \in S_Y$ there must correspond a path with embedded cycles in G_d formed with certain states and a trace $s'_Y = P_{oo'}(s_Y)$ formed with the events of a path with embedded indeterminate cycles in G'_d. Therefore, since, by assumption, there is no $s_Y \in L'_{\text{test}}$ such that $s' \in \overline{P_{oo'}(s_Y)}$, and only paths with embedded cycles formed with normal states of G_d whose events form a trace s_N such that $s' \in \overline{P_{oo'}(s_N)}$ can be found in G'_{test}, it is not possible to have s' associated with a prime path of G'_d whose unique cycle is indeterminate, which contradicts the assumption that L is not diagnosable with respect to P'_o and Σ_f. Similar argument can be used to prove that when condition (ii) holds true, there is also a contradiction. □

Remark 7 Note that if s' is associated with a cycle of uncertain states that do not form an indeterminate cycle, then there will not exist s_Y such that $s' \in \overline{P_{oo'}(s_Y)}$, but only s_N such that $s' \in \overline{P_{oo'}(s_N)}$. The reason for that is the assumption that L is diagnosable with respect to P_o and Σ_f.

Theorem 5 above establishes that for any trace s' formed with events of a prime path of G'_d whose unique cycle is indeterminate and observed, it is always possible to find, at least, a pair of traces $(s_Y, s_N) \in S_Y \times S_N$ such that $s' \in \overline{P_{oo'}(s_Y)}$ and $s' \in \overline{P_{oo'}(s_N)}$. It is worth remarking that the reverse is not necessarily true, *i.e.*, we may not state that for any trace $s_Y \in S_Y$, it is possible to find a trace s' formed with events of a prime path of G'_d such that, for a trace $v \in \overline{s_Y}$, $s' = P_{oo'}(v)$. The same conclusion can be drawn for traces $s_N \in S_N$. In spite of that, the following result can be stated.

Proposition 3 *For any trace $s \in S_Y \cup S_N$, there exists a trace $u \in \bar{s}$ such that $P_{oo'}(u) = s'$, where s' is a trace formed with the events of a prime path of G'_d (formed from a non-hidden cycle) whose unique embedded cycle satisfies one of the following conditions:* (i) *it is indeterminate;* (ii) *it is formed with normal states;* (iii) *it is formed with uncertain states that do not create an indeterminate cycle.*

Proof For a state to be revisited in a path of G'_{test}, a state of a path of G'_d has to be revisited through the events of the corresponding path of G'_{test} that belong to Σ'_o. This implies that the projection of a prime path of G'_{test} is not necessarily a prime path of G'_d, but it is always a path with embedded cycles. Nevertheless, any trace $s \in S_Y \cup S_N$ has a prefix u such that $P_{oo'}(u) = s'$, where s' is a trace formed with events of a prime path of G'_d, which not necessarily contains an indeterminate cycle. This is so because a diagnoser may cycle over normal states or uncertain states that do not form an indeterminate cycle before it starts to cycle over uncertain states of an indeterminate cycle. Notice that s' cannot be a trace associated with a prime path formed from a cycle of certain states since it is never possible for a diagnoser to go from certain to uncertain states. □

The main implication of Theorem 5 and Proposition 3 is that in order to find an innovate event set for Σ'_o that deals with indeterminate observed cycles in G'_d, it is

first necessary to find traces $s_Y \in S_Y$ and $s_N \in S_N$ for which there exist a trace s' formed with the events of a prime path of G'_d whose unique cycle is indeterminate and observed and satisfies $s' \in \overline{P_{oo'}(s_Y)}$ and $s' \in \overline{P_{oo'}(s_N)}$. Let $x^\star_{t,Y} = (x'^\star_d, x^\star_d)$ denote the unique revisited state of a Y-prime path of G'_{test} and write s_Y as $s_Y = u_Y v_Y$, where u_Y and v_Y, are such that $x^\star_{t,Y} = f_t(x_{0_t}, u_Y)$ and $f_t(x^\star_{t,Y}, v_Y) = x^\star_{t,Y}$. Therefore, in order for the Y-prime path formed with the events of s_Y be associated with an indeterminate observed cycle of G'_d, at least one event of v_Y must belong to Σ'_o. The same condition applies to traces $s_N \in S_N$.

The following result shows how to use the events of s_Y and s_N that belong to $\Sigma_o \setminus \Sigma'_o$ to form innovative event sets for Σ'_o.

Proposition 4 *Assume that language L is not diagnosable with respect to P'_o and Σ_f and let $\Sigma''_o = \Sigma'_o \cup \Sigma_{ies}$, $\Sigma_{ies} \subseteq \Sigma_o \setminus \Sigma'_o$ and let G''_d denote the partial diagnoser for L assuming Σ''_o as the observable event set. In addition, let $(s_Y, s_N) \in S_Y \times S_N$ satisfying $P_{oo'}(s_Y) = P_{oo'}(s_N)$, and consider a trace s' associated with a prime path of G'_d whose unique cycle is indeterminate (but not hidden) and $s' \in \overline{P_{oo'}(s_Y)}$ and $s' \in \overline{P_{oo'}(s_N)}$. A necessary condition for s' not to be a trace associated with a prime path of G''_d, whose unique cycle is indeterminate, is that $\Sigma_{ies} \cap [(\Sigma_{s_Y} \cup \Sigma_{s_N}) \setminus \Sigma'_o] \neq \emptyset$, where Σ_{s_Y} and Σ_{s_N} denote, respectively, the sets formed with the events of traces s_Y and s_N, respectively.*

Proof Let $\Sigma_{ies} = \{\sigma \in \Sigma_o : \sigma \notin (\Sigma_{s_Y} \cup \Sigma_{s_N}) \setminus \Sigma'_o\}$ and assume that all traces $s'' = P_{oo''}[P^{-1}_{oo'}(s') \cap L_d]$ are associated with paths of G''_d whose embedded cycles are not indeterminate. However, $P^{-1}_{oo'}(s') \cap L_d \supseteq \{s_Y, s_N\}$, then $P_{oo''}(s_Y) = P_{oo''}(s_N) = s'$, which contradicts the assumption that s'' is not associated with a path of G''_d with embedded indeterminate cycles. □

Remark 8 It is worth remarking that the condition given in Proposition 4 is only necessary, since if a common event of s_Y and s_N is included in Σ_{ies}, it may still be possible that $P_{oo''}(s_Y) = P_{oo''}(s_N) = s''$, which implies that s'' may also be associated with a path with embedded indeterminate cycles.

According to Proposition 4, a necessary condition for a pair of traces $(s_Y, s_N) \in S_Y \times S_N$ that satisfy $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'$, not to lead to paths with embedded indeterminate cycles in G''_d, is that, at least, one event of s_Y or one event of s_N be in Σ_{ies}. Therefore, this requirement must be satisfied for all pairs of traces (s_Y, s_N) in G'_{test} that lead to some trace s' associated with a prime path whose unique cycle is indeterminate and observed. Based on this fact, we propose Algorithm 3 that returns a set Σ^o_{ies} whose elements are sets formed with events of $\Sigma_o \setminus \Sigma'_o$ that must be added to Σ'_o in order to create new candidates for minimal diagnosis bases. A requirement to perform the algorithm is that all prime paths of G'_d and G'_{test} have already been calculated.

Steps 1 and 2 of Algorithm 3 identifies all prime paths of G'_d and G'_{test} whose unique cycle is indeterminate and observed. Step 3 forms sets whose traces have as projections the traces belonging to S'_d, therefore, identifying the pairs that satisfy the conditions of Theorem 5. In steps 4 and 5, innovative event sets are formed so as to satisfy the conditions imposed by Proposition 4 for each pair of traces s_Y and s_N such that $P_{oo''}(s_Y) = P_{oo''}(s_N) = s'$ for each $s' \in S'_d$. Step 6 considers the construction of

Algorithm 3 (Computation of the innovative event sets associated with indeterminate observed cycles of G'_d)

STEP 1 Form set

$$S'_d = \{s' \in {\Sigma'_o}^* : s' \text{ is a trace associated with a prime path of } G'_d \text{ formed from an indeterminate observed cycle}\}.$$

If $S'_d = \emptyset$ then $\Sigma^o_{ies} = \{\emptyset\}$ and stop. Otherwise, go to Step 2.

STEP 2 Form sets S_Y and S_N according to Eqs. 17 and 18, respectively. For all prime paths associated with the traces in S_Y and S_N, identify the unique revisited states $x^\star_{t,Y}$ and $x^\star_{t,N}$, respectively, and the corresponding traces $s_Y = u_Y v_Y$ and $s_N = u_N v_N$ such that $x^\star_{t,Y} = f_t(x_{0_t}, u_Y)$ and $f_t(x^\star_{t,Y}, v_Y) = x^\star_{t,Y}$ and $x^\star_{t,N} = f_t(x_{0_t}, u_N)$ and $f_t(x^\star_{t,N}, v_N) = x^\star_{t,N}$. Form the following sets:

$$S^o_Y = \{s_Y = u_Y v_Y \in S_Y : v_Y \text{ has at least one event in } \Sigma'_o\},$$

$$S^o_N = \{s_N = u_N v_N \in S_N : v_N \text{ has at least one event in } \Sigma'_o\}.$$

STEP 3 Let $S'_d = \{s'_1, s'_2, \ldots, s'_p\}$, where $p = |S'_d|$. For each $s'_i \in S'_d$, $i = 1, \ldots, p$, form the following sets:

$$S^o_{Y,i} = \{s_Y \in S^o_Y : s'_i \in \overline{P_{oo'}(s_Y)}\},$$

$$S^o_{N,i} = \{s_N \in S^o_N : s'_i \in \overline{P_{oo'}(s_N)}\}.$$

STEP 4 For each trace $s^k_{Yi} \in S^o_{Y,i}$ form a set $\Sigma^k_{Y,i}$ with the events of s^k_{Yi} that belong to $\Sigma_o \setminus \Sigma'_o$. For each trace $s^l_{Ni} \in S^o_{N,i}$ form a set $\Sigma^l_{N,i}$ with the events of s^l_{Ni} that belong to $\Sigma_o \setminus \Sigma'_o$.

STEP 5 Let $l_{Yi} = |S^o_{Y,i}|$ and $l_{Ni} = |S^o_{N,i}|$. For $i = 1, \ldots, p$, compute:

$$\Sigma^o_{ies,Yi} = \begin{cases} \{\emptyset\}, & \text{if } l_{Yi} = 1 \text{ and } \Sigma_{Y,i} = \emptyset \\ 2_1^{\Sigma_{Y,i}}, & \text{if } l_{Yi} = 1 \text{ and } \Sigma_{Y,i} \neq \emptyset \\ \Sigma^1_{Y,i} \dot{\times} \Sigma^2_{Y,i} \dot{\times} \ldots \dot{\times} \Sigma^{l_{Yi}}_{Y,i}, & \text{if } l_{Yi} > 1, \end{cases}$$

$$\Sigma^o_{ies,Ni} = \begin{cases} \{\emptyset\}, & \text{if } l_{Ni} = 1 \text{ and } \Sigma_{N,i} = \emptyset \\ 2_1^{\Sigma_{N,i}}, & \text{if } l_{Ni} = 1 \text{ and } \Sigma_{N,i} \neq \emptyset \\ \Sigma^1_{N,i} \dot{\times} \Sigma^2_{N,i} \dot{\times} \ldots \dot{\times} \Sigma^{l_{Ni}}_{N,i}, & \text{if } l_{Ni} > 1, \end{cases}$$

$$\Sigma^o_{ies,i} = \Sigma^o_{ies,Yi} \cup \Sigma^o_{ies,Ni}.$$

STEP 6 Compute $\Sigma^o_{ies} = \Sigma^o_{ies,1} \dot{\times} \Sigma^o_{ies,2} \dot{\times} \ldots \dot{\times} \Sigma^o_{ies,p}$.

STEP 7 Remove from Σ^o_{ies}, all event sets $\Sigma' \in \Sigma^o_{ies}$ for which there exists another set $\Sigma'' \in \Sigma^o_{ies}$ such that $\Sigma' \supseteq \Sigma''$.

innovative event sets for each pair or traces obtained in step 3 and form a set whose elements have at least one element of each set computed in step 5. Finally, step 7 removes all supersets of innovative event sets that appear in Σ^o_{ies}, since only minimal diagnosis bases are being sought.

The example below illustrates the application of Algorithm 3.

Example 3 Consider automaton G whose state transition diagram is shown in Fig. 6 and let $\Sigma_o = \{a,b,c,d\}$, $\Sigma_{uo} = \{\sigma, \sigma_f\}$ and $\Sigma_f = \{\sigma_f\}$ be, respectively, the observable, the unobservable and fault event sets for G. The language L generated by G is clearly diagnosable with respect to $P_o : E^* \to \Sigma_o^*$ and Σ_f, since the diagnoser G_d, shown in Fig. 7, has no indeterminate cycles.

Let us assume that we are interested in finding all minimal diagnosis bases for L. In order to achieve this goal, the first step is to find the set of elementary diagnosing event sets. Using Algorithm 1, the following set is obtained:

$$\Sigma_{\text{edes}} = \{\{a,c\}\}.$$

Therefore set $\Sigma'_o = \{a,c\}$ is the unique set to be considered. Figure 8 shows the corresponding partial diagnoser G'_d, which possesses both indeterminate observed and indeterminate hidden cycles. Therefore, L is not diagnosable with respect to P'_o and Σ_f, which means that $\Sigma'_o = \{a,c\}$ is not a minimal diagnosis basis.

Let us address, now, the problem of obtaining a new set $\Sigma''_o = \Sigma'_o \cup \Sigma_{ies}$ of least possible cardinality with the view to removing the indeterminate observed cycles that exists in G'_d. Following Algorithm 3, the first step is to form the tree of Fig. 9 and, in the sequel, the set S'_d, whose elements are the traces formed with the events of prime paths of G'_d associated with indeterminate observed cycles. This is done by searching on the tree shown in Fig. 9 for the leaves labeled with uncertain states of G'_d that form indeterminate observed cycles. It is clear from Fig. 9 that the leaves labeled as YN_1 and YN_2, corresponding, respectively, to states $\{9N,10Y,11Y\}$ and $\{9N,10Y\}$ of G'_d, are the only ones whose unique cycles of their prime paths are indeterminate and observed. Therefore, set S'_d will be given as:

$$S'_d = \{s'_1,s'_2\} = \{accc,ccc\}.$$

Since S'_d has two elements, we can move to Step 2 of Algorithm 3. In order to obtain the sets S^o_Y and S^o_N it is necessary to build a tree for G'_{test}, form sets S_Y and S_N,

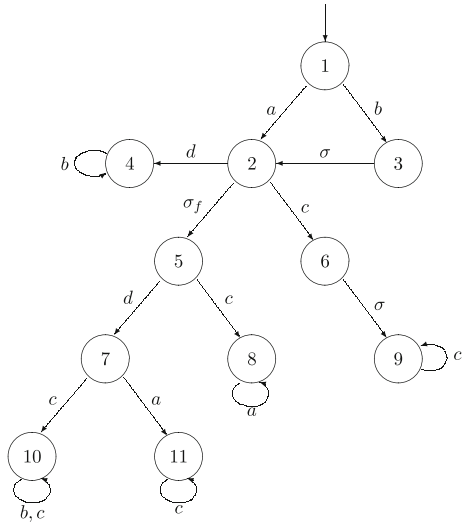


Fig. 6 Automaton G for the illustration of Algorithms 3 and 4

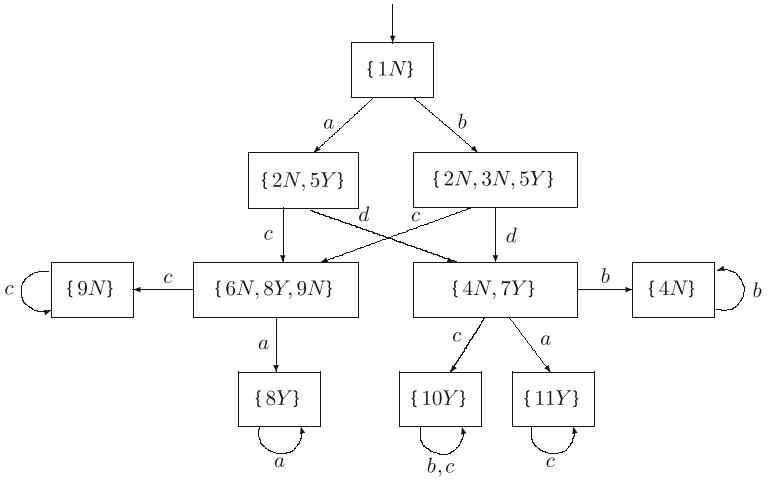


Fig. 7 Centralized diagnoser G_d for automaton G of Fig. 6

and identify all traces of S_Y and S_N associated with indeterminate observed cycles of G'_d. Using the tree of Fig. 11—which was built for automaton G'_{test} depicted in Fig. 10—we find the following traces of S_Y associated with indeterminate observed cycles: $s_{Y,1} = adccc$, $s_{Y,2} = bdaccc$, and $s_{Y,3} = bdccc$. Therefore,

$$S_Y^o = \{adccc, bdaccc, bdccc\}.$$

The traces of S_N associated with indeterminate cycles of G'_d are also obtained using the tree of Fig. 11, being given as: $s_{N,1} = accc$ and $s_{N,2} = bccc$. Therefore, set S_N^o is given as:

$$S_N^o = \{bccc, accc\}.$$

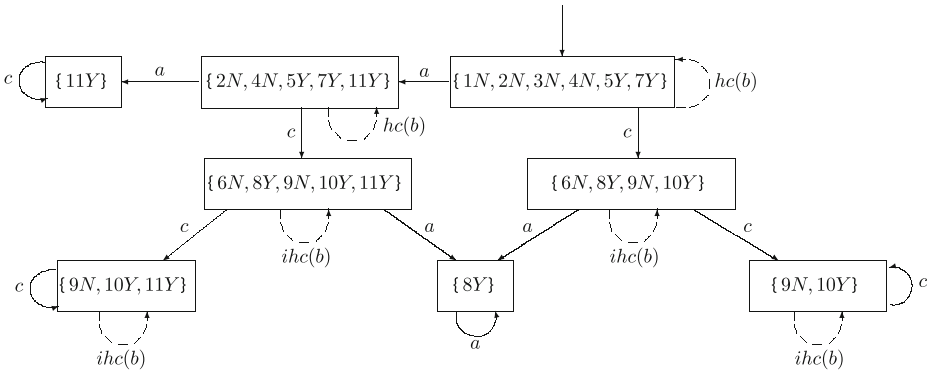
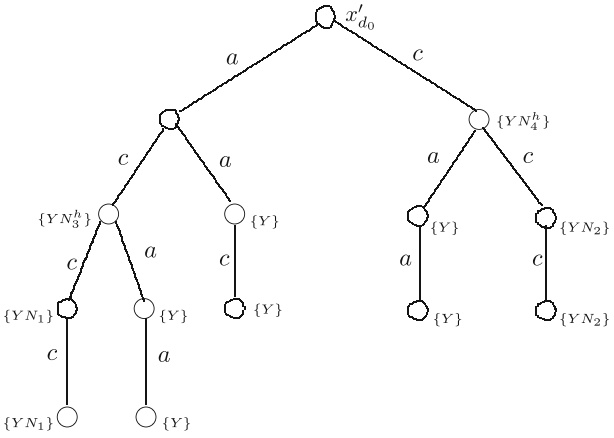


Fig. 8 Partial diagnoser G'_d for automaton G of Fig. 6 assuming $\Sigma'_o = \{a, c\}$

Fig. 9 Tree for G'_d



Following, now, Step 3 of Algorithm 3, the following sets are formed:

$$S^o_{Y,1} = \{adccc, bdaccc\},\ S^o_{Y,2} = \{bdccc\},\ S^o_{N,1} = \{accc\},\ \text{and}\ S^o_{N,2} = \{bccc\}.$$

Notice that, in this particular example, the projections of the traces in $S^o_{Y,1}$, $S^o_{Y,2}$, $S^o_{N,1}$, and $S^o_{N,2}$ are equal to traces s'_1 and s'_2 of S'_d.

Having formed sets $S^o_{Y,1}$, $S^o_{Y,2}$, $S^o_{N,1}$, and $S^o_{N,2}$, the next step is to obtain the set of events of traces $s^i_Y \in S_{Y,i}$ and $s^i_N \in S_{N,i}$ that are not in Σ'_o. The following sets are then formed:

$$\Sigma^1_{Y,1} = \{d\},\ \Sigma^2_{Y,1} = \{b, d\},\ \Sigma^1_{Y,2} = \{b, d\},\ \Sigma^1_{N,1} = \emptyset\ \text{and}\ \Sigma^1_{N,2} = \{b\}.$$

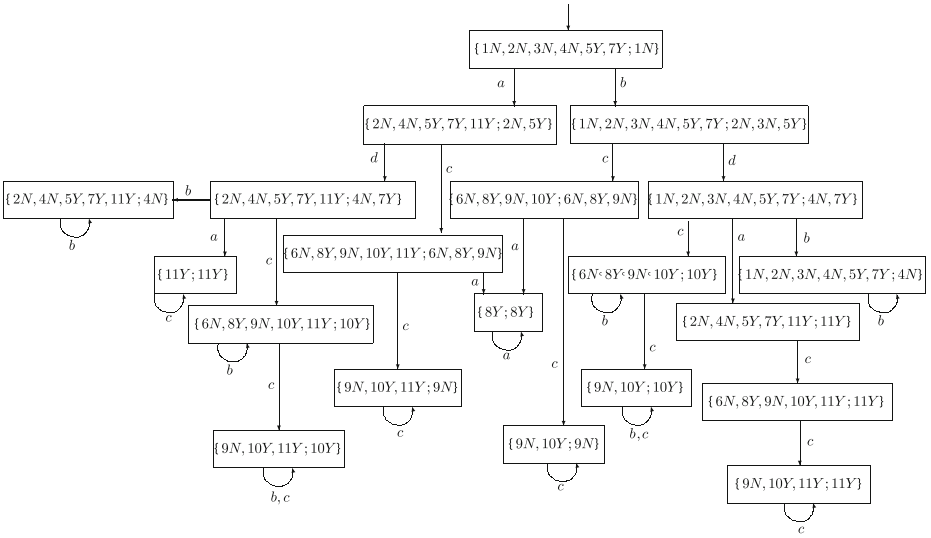


Fig. 10 Automaton $G'_{\text{test}} = G'_d \| G_d$.

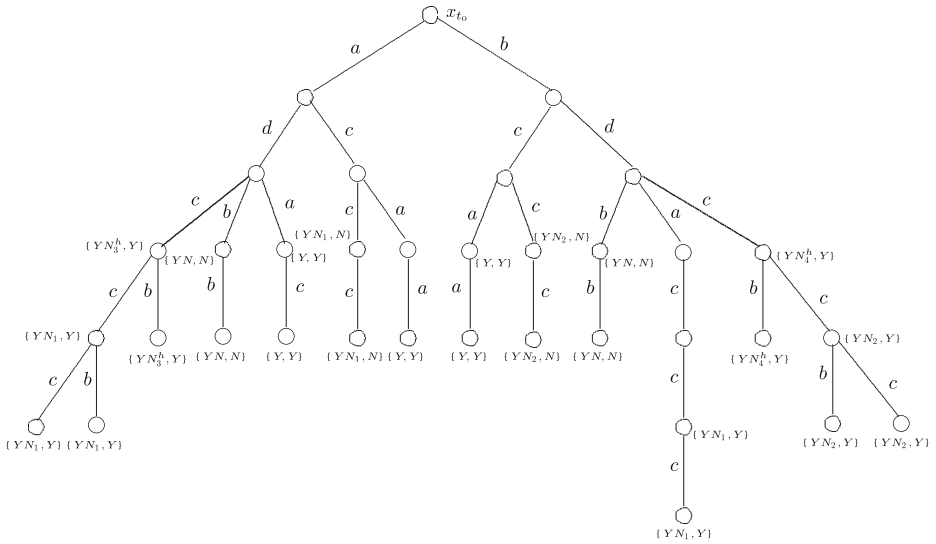


Fig. 11 Tree for G'_{test}

Proceeding now according to Step 6, we obtain:

$$\Sigma^o_{ies,Y1} = \Sigma^1_{Y,1} \dot{\times} \Sigma^2_{Y,1} = \{\{b,d\},\{d\}\},\ \Sigma_{ies,Y2} = \{\{b\},\{d\}\},$$
$$\Sigma^o_{ies,N1} = \{\emptyset\},\ \Sigma_{ies,N2} = \{\{b\}\},$$
$$\Sigma^o_{ies,1} = \Sigma^o_{ies,Y1} \cup \Sigma^o_{ies,N1} = \{\{b,d\},\{d\}\}$$
$$\Sigma^o_{ies,2} = \Sigma^o_{ies,Y2} \cup \Sigma^o_{ies,N2} = \{\{b\},\{d\}\}.$$

Therefore, the innovative event sets for the indeterminate observed cycles of G'_d are given as:

$$\Sigma^o_{ies} = \Sigma_{ies,1} \dot{\times} \Sigma_{ies,2} = \{\{b,d\},\{d\}\}.$$

Finally, since $\{d\} \subset \{b,d\}$, then $\{b,d\}$ must be removed from Σ^o_{ies}. Therefore,

$$\Sigma^o_{ies} = \{\{d\}\}.$$

5.3 Dealing with indeterminate hidden cycles of G'_d

According to conditions HC1. and HC2., the existence of indeterminate hidden cycles in G'_d implies that there exist at least one arbitrarily long trace $s_Y \in L_d$ associated with a path with embedded cycles formed with certain states of G_d, and one trace of bounded length $s_N \in L_d$, that takes G_d from its initial state to either a normal or an uncertain state, satisfying $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'$, where s' is a bounded length trace. In addition, notice that, since $L'_{\text{test}} = L_d$, the cycles that are hidden in G'_d have corresponding cycles in G'_{test} whose states all have the same first component, namely, the state of G'_d that contains the hidden cycle. Moreover, the events that label the transitions between these states all belong to $\Sigma_o \setminus \Sigma'_o$. This

suggests that it is possible to establish a connection between Y-prime paths of G'_{test} and paths of G'_d with embedded indeterminate hidden cycles.

Let us define the following sets:

$$S'_t = \{s : s \text{ is a trace associated with a prime path of } G'_{\text{test}}\}, \tag{19}$$

$$S_{YY} = \{s \in S'_t : (\text{there exists a leaf labeled as } (x'_d, x_d), \text{ where } x'_d \text{ and } x_d \text{ are both certain})[f_t(x_{t_0},s) = (x'_d,x_d)]\}, \tag{20}$$

$$S_{NN} = \{s \in S'_t : (\text{there exists a leaf labeled as } (x'_d, x_d), \text{ where } x'_d \text{ and } x_d \text{ are both normal})[f_t(x_{t_0},s) = (x'_d,x_d)]\}. \tag{21}$$

Notice that S_{YY} (respectively S_{NN}) is formed with prime paths whose unique cyclic path is formed with states whose components are both certain (respectively both normal).

Theorem 6 *Let $x^\star_t = (x'^\star_d, x^\star_d)$ be the unique revisited state of a Y-prime path of G'_{test}, and let s_Y denote the trace formed with the events of this Y-prime path. In addition, assume that $s_Y = uv$, where $v \in (\Sigma_o \setminus \Sigma'_o)^+$, $x^\star_t = f_t(x_{0_t}, u)$, and $f_t(x^\star_t, v) = x^\star_t$. Then $s' = P_{oo'}(s_Y)$ is a trace formed with events of a path with embedded indeterminate hidden cycles of G'_d in state $x'^\star_d = f'_d(x'_{0_d}, s')$. In addition, let*

$$S^h_Y(s') = \{s_Y \in S^h_Y : P_{oo'}(s_Y) = s'\}, \tag{22}$$

where

$$S^h_Y = \{s_Y \in S_Y : (s_Y = uv \in \Sigma^*_o)(v \in (\Sigma_o \setminus \Sigma'_o)^+)\}, \tag{23}$$

and define the following set:

$$S^h_N(s') = S^{YN}_Y(s') \cup S^{YN}_{YY} \cup S_N \cup S_{NN}, \tag{24}$$

where S_N and S_{NN} are defined in Eqs. 18 and 21, respectively,

$$S^{YN}_{YY} = S^{YN}_{YY,1} \setminus S^{YN}_{YY,2}, \tag{25}$$

where

$$S^{YN}_{YY,1} = \left\{s \in \overline{S_{YY}} : (\exists x_d \textit{ uncertain})[f_t(x_{t_0},s) = (x'_d,x_d)]\right\}, \tag{26}$$

$$S^{YN}_{YY,2} = \left\{s \in S^{YN}_{YY,1} : \left(\exists s_{\max} \in S^{YN}_{YY,1}\right)[(s \in \overline{s_{\max}}) \wedge (s \neq s_{\max})]\right\}, \tag{27}$$

with S_{YY} defined in Eq. 20, and

$$S^{YN}_Y(s') = S^{YN}_{Y,1}(s') \setminus S^{YN}_{Y,2}(s'), \tag{28}$$

where

$$S^{YN}_{Y,1}(s') = \left\{s \in \overline{S_Y(s')} : (\exists x_d \textit{ uncertain})[f_t(x_{t_0},s) = (x'_d,x_d)]\right\}, \tag{29}$$

$$S^{YN}_{Y,2}(s') = \left\{s \in S^{YN}_{Y,1}(s') : \left(\exists s_{\max} \in S^{YN}_{Y,1}(s')\right)[(s \in \overline{s_{\max}}) \wedge (s \neq s_{\max})]\right\}, \tag{30}$$

with

$$S_Y(s') = S^h_Y \setminus S^h_Y(s'). \tag{31}$$

Then, there exists at least one trace $s_P \in S_N^h(s')$ that has a prefix s_N ($s_N \in \overline{s_P}$) such that $P_{oo'}(s_N) = s'$ and $f_d(x_{0_d}, s_P) = x_d^{\#}$, where $x_d^{\#}$ is either a normal state of G_d or an uncertain state of G_d that is not a state of an indeterminate cycle.

Proof Let us consider the first part of the theorem. Since $G'_d = Obs(G_d, \Sigma'_o)$ (up to state renaming) and $G'_{\text{test}} = G'_d \| G_d$, it is not hard to check that for all state $x_t = (x'_d, x_d) \in X_t$, then $x_d \subseteq x'_d$. In addition, since by assumption $v \in (\Sigma_o \setminus \Sigma'_o)^+$ and satisfies $f_t(x_t^\star, v) = x_t^\star$, then the first components of all states reached after u are all equal to $x'_d = f'_d(x'_{0_d}, s')$. Moreover, since $L'_{\text{test}} = L_d$ and the second components of the states of the Y-prime path reached after u are all subsets of x'_d, we may conclude that x'_d possesses an indeterminate hidden cycle, which proves the first part of the lemma.

Consider, now, the second part of the theorem. Let $x_d'^\star$ be an uncertain state of G'_d that contains an indeterminate hidden cycle, and assume that there does not exist any trace $s_P \in S_N^h(s')$ that has a prefix s_N satisfying $P_{oo'}(s_N) = s'$. Notice that set $S_N^h(s')$ is formed with the longest prefixes of each trace of S_{YY} and $S_Y \setminus \{s_Y\}$ that lead to uncertain states of G_d not belonging to an indeterminate cycle and with the traces of S_N and S_{NN} which, by definition, lead to normal states of G_d. It is known that in order for a state $x'_d \in X'_d$ to have an indeterminate hidden cycle, there must exist traces $\tilde{s}_Y, \tilde{s}_N \in L'_{\text{test}}$, where $\tilde{s}_Y$ is formed with the events of a path with embedded cycles of certain states of G_d, and $\tilde{s}_N$ with bounded length, such that $\tilde{x}_d = f_d(x_{0_d}, \tilde{s}_N)$ is either a normal or an uncertain state of G_d, and $P_{oo'}(\tilde{s}_N) = P_{oo'}(\tilde{s}_Y) = \tilde{s}'$, for some trace $\tilde{s}' \in L'_d$. Since $L'_{\text{test}} = L_d$, and by assumption, there exists no trace s_P that takes G_d from the initial state to either a normal or an uncertain state, we may conclude that $x_d'^\star = f'_d(x'_{0_d}, s')$ is a certain state of G'_d, which contradicts the fact that $x_d'^\star$ is an uncertain state. □

Let us now address the problem of finding an innovative event set Σ_{ies} in such a way that the hidden cycles that appear in G'_d will not appear in G''_d. As in the case of indeterminate observed cycles, the idea here is to include events of $\Sigma_o \setminus \Sigma'_o$ with the view to making $P_{oo''}(s_Y) \neq P_{oo''}(s_N)$ for all $s_Y, s_N \in L_d$ that satisfy conditions HC1. and HC2.

Proposition 5 *Assume that L is not diagnosable with respect to P'_o and Σ_f and let $\Sigma''_o = \Sigma'_o \cup \Sigma_{ies}$, $\Sigma_{ies} \subseteq \Sigma_o \setminus \Sigma'_o$. Let (s_Y, s_N) be a pair of traces in L'_{test} where s_Y is a trace associated with a Y-prime path of G'_{test} that corresponds to a path with an embedded indeterminate hidden cycle in G'_d, and s_N is a prefix of a trace $s_P \in S_N^h(s')$, where $s' = P_{oo'}(s_Y)$, whose last event $s_{N_f} \in \Sigma'_o$. A necessary condition for the path with embedded indeterminate hidden cycle associated with s' not to be a path of G''_d with embedded indeterminate hidden cycles, is that $\Sigma_{ies} \cap [(\Sigma_{s_Y} \cup \Sigma_{s_N}) \setminus \Sigma'_o] \neq \emptyset$, where Σ_{s_Y} and Σ_{s_N} denote, respectively, the sets formed with the events of traces s_Y and s_N.*

Proof Let $\Sigma_{ies} = \{\sigma \in \Sigma_o : \sigma \notin (\Sigma_{s_Y} \cup \Sigma_{s_N}) \setminus \Sigma'_o\}$, and assume that all traces $s'' = P_{oo''}[P_{oo'}^{-1}(s') \cap L_d]$ are associated with paths of G''_d that possess no indeterminate hidden cycles. However, since $P_{oo'}^{-1}(s') \cap L_d \supseteq \{s_Y, s_N, s_P\}$, then $P_{oo''}(s_Y) = P_{oo''}(s_N) = P_{oo''}(s_P) = s'$, which contradicts the assumption that s'' is not associated with a path of G''_d with embedded indeterminate hidden cycles. □

Remark 9

(a) The reader could argue that instead of restricting the events of Σ_{ies} to $(\Sigma_{s_Y} \cup \Sigma_{s_N}) \setminus \Sigma'_o$, we could select events from $(\Sigma_{s_Y} \cup \Sigma_{s_P}) \setminus \Sigma'_o$ to form Σ_{ies}. Notice that, if, for instance, $\Sigma_{ies} = \{\sigma\}$, where $\sigma \in \Sigma_{s_P}$ but $\sigma \notin \Sigma_{s_N}$ and $\sigma \notin \Sigma_{s_Y}$, then, although $P_{oo''}(s_P) \neq s'$, we still would have $P_{oo''}(s_Y) = P_{oo''}(s_N) = s'$ since $P^{-1}_{oo'}(s') \cap L_d \supseteq \{s_Y, s_N, s_P\}$, which again would lead to an indeterminate hidden cycle in G''_d.

(b) Although the condition imposed by Proposition 5 is only necessary, it may become sufficient if additional assumptions are made. For example, let $\Sigma_{ies} = \{\sigma\}$, where $\sigma \in s_N$ and $\sigma \notin s_Y$, and write $\hat{s}_N = s_N\hat{w} = t\sigma w\hat{w}$. It is not difficult to see that: (i) $P_{oo'}(s_Y) = P_{oo'}(s_N) = P_{oo'}(t)P_{oo'}(w)$; (ii) $P_{oo''}(s_Y) = P_{oo'}(s_Y) = P_{oo'}(t)P_{oo'}(w)$ and; (iii) $P_{oo''}(s_N) = P_{oo''}(t)\sigma P_{oo''}(w)$. As a consequence, even in the case when $\sigma \notin t$ or $\sigma \notin w$, $P_{oo''}(s_Y) \neq P_{oo''}(s_N)$ and $P_{oo''}(s_Y) \neq P_{oo''}(\hat{s}_N)$. In addition, since $\sigma \notin s_Y$ and $P_{oo''}(t) \neq \epsilon$, it is not possible to find a prefix of s_N with the same projection over $\Sigma_o''^*$ as s_Y, which implies that $P_{oo''}[P^{-1}_{oo'}(s') \cap L_d]$ does not lead to any indeterminate cycles (observed or hidden).

According to Proposition 5, for a pair of traces (s_Y, s_P), where s_Y is a trace formed with the events of a Y-prime path of G'_{test} associated with a path with an embedded indeterminate hidden cycle in G'_d, and $s_P \in S^h_N(s')$ and has a prefix s_N whose last event $s_{N_f} \in \Sigma'_o$ and satisfy $P_{oo'}(s_Y) = P_{oo'}(s_N) = s'$, not to lead to paths with embedded indeterminate hidden cycles in G''_d, it is necessary to include in Σ_{ies}, either at least one event of s_Y or at least one event of s_N that belongs to $\Sigma_o \setminus \Sigma'_o$. Furthermore, this requirement must be satisfied for all pairs of traces (s_Y, s_N) in G'_{test} that satisfy the above conditions. Algorithm 4 uses these facts in order to build a set of innovative event sets with the view to avoiding indeterminate hidden cycles in G''_d. As for Algorithm 3, it will be assumed that all prime paths of G'_d and G'_{test} have already been calculated.

Steps 1 to 5 of Algorithm 4 are derived directly from Theorem 6 and are intended to find all events that belong to $\Sigma_o \setminus \Sigma'_o$ in order to prevent s'_i from being a trace associated with an indeterminate hidden cycle when the observable event set becomes $\Sigma''_o = \Sigma'_o \cup \Sigma_{ies}$, $\Sigma_{ies} \subseteq (\Sigma_o \setminus \Sigma'_o)$. Step 6 considers, as in Algorithm 3, the construction of innovative event sets for each pair of traces obtained in steps 3 and 4 and form a set whose elements have at least one event of each set computed in step 6. Finally, step 7 removes all supersets of innovative event sets that appear in Σ^h_{ies}, since only minimal diagnosis bases are being sought.

Algorithm 4 will be illustrated by means of the following example.

Example 4 Let us consider again automaton G whose state transition diagram is depicted in Fig. 6. We make here the same assumptions regarding the observable, unobservable and fault event sets of G, *i.e.*, $\Sigma_o = \{a,b,c,d\}$, $\Sigma_{uo} = \{\sigma, \sigma_f\}$ and $\Sigma_f = \{\sigma_f\}$, respectively. As shown in Example 3, $\Sigma'_o = \{a,c\}$ is not a minimal diagnosis basis for L, which imposes the need to find innovative event sets to remove the indeterminate (observed and hidden) cycles that exist in G'_d. The indeterminate observed cycles were considered in Example 3. We will now consider the hidden cycles of G'_d.

According to Algorithm 4, the first step is to form set S^h_Y whose elements are the traces formed with the events of a Y-prime path of G'_{test} associated with the

Algorithm 4 (Computation of the innovative event sets associated with indeterminate hidden cycles of G'_d)

STEP 1 Identify for each Y-prime path of G'_{test}, the unique revisited states $x^\star_{t,Y}$ and the corresponding trace $s_Y = u_Y v_Y$ such that $x^\star_{t,Y} = f_t(x_{0_t}, u_Y)$ and $f_t(x^\star_{t,Y}, v_Y) = x^\star_{t,Y}$, and form the following sets:

$$S^h_Y = \{s_Y = u_Y v_Y \in \Sigma^\star_o : v_Y \in (\Sigma_o \setminus \Sigma'_o)^+\},$$

and compute $P_{oo'}(S^h_Y)$. Define $p = |P_{oo'}(S^h_Y)|$ $(p \leq |P_{oo'}(S^h_Y)|)$, and write $P_{oo'}(S^h_Y)$ as

$$P_{oo'}(S^h_Y) = \{s'_1, s'_2, \ldots, s'_p\}.$$

STEP 2 Form sets S_N, S_{NN} according to Eqs. 18 and 21, respectively, and set S^{YN}_{YY} according to Eqs. 20, 25, 26 and 27.

STEP 3 For each $s'_i \in P_{oo'}(S^h_Y)$, $i = 1, \ldots, p$, form set $S^h_{Y,i} = \{s_Y \in S^h_Y : P_{oo'}(s_Y) = s'_i\}$.

STEP 4 For each $s'_i \in P_{oo'}(S^h_Y)$, $i = 1, \ldots, p$, form set $S^h_{N,i}$ as follows:

- Form set $S^h_N(s'_i)$ according to Eqs. 24, 28, 29, and 30, with s' replaced with s'_i, and where in Eq. 29, $S_Y(s'_i) = S^h_Y \setminus S^h_{Y,i}$.
- Form set

$$S^h_{N,i} = \{s_N \in \overline{S^h_N(s'_i)} : (\exists s_{N_f} \in \Sigma'_o)[P_{oo'}(s_N) = s'_i]\}, \tag{32}$$

where s_{N_f} denotes the last event of s_N.

STEP 5 Form sets $\Sigma^k_{Y,i}$ and $\Sigma^l_{N,i}$ as follows:

- For each $s^k_{Y,i} \in S^h_{Y,i}$ form a set $\Sigma^k_{Y,i}$ with the events of $s^k_{Y,i}$ that belong to $\Sigma_o \setminus \Sigma'_o$.
- For each $s^l_{N,i} \in S^h_{N,i}$ form a set $\Sigma^l_{N,i}$ with the events of $s^l_{N,i}$ that belong to $\Sigma_o \setminus \Sigma'_o$.

STEP 6 Let $l_{Yi} = |S^h_{Y,i}|$ and $l_{Ni} = |S^h_{N,i}|$. For $i = 1, \ldots, p$, compute:

$$\Sigma^h_{ies,Yi} = \begin{cases} \{\emptyset\}, & \text{if } l_{Yi} = 1 \text{ and } \Sigma_{Y,i} = \emptyset \\ 2_1^{\Sigma_{Y,i}}, & \text{if } l_{Yi} = 1 \text{ and } \Sigma_{Y,i} \neq \emptyset \\ \Sigma^1_{Y,i} \dot{\times} \Sigma^2_{Y,i} \dot{\times} \ldots \dot{\times} \Sigma^{l_{Yi}}_{Y,i}, & \text{if } l_{Yi} > 1, \end{cases}$$

$$\Sigma^h_{ies,Ni} = \begin{cases} \{\emptyset\}, & \text{if } l_{Ni} = 1 \text{ and } \Sigma_{N,i} = \emptyset \\ 2_1^{\Sigma_{N,i}}, & \text{if } l_{Ni} = 1 \text{ and } \Sigma_{N,i} \neq \emptyset \\ \Sigma^1_{N,i} \dot{\times} \Sigma^2_{N,i} \dot{\times} \ldots \dot{\times} \Sigma^{l_{Ni}}_{N,i}, & \text{if } l_{Ni} > 1, \end{cases}$$

$$\Sigma^h_{ies,i} = \Sigma^h_{ies,Yi} \cup \Sigma^h_{ies,Ni}.$$

STEP 7 Compute $\Sigma^h_{ies} = \Sigma^h_{ies,1} \dot{\times} \Sigma^h_{ies,2} \dot{\times} \ldots \dot{\times} \Sigma^h_{ies,p}$.

STEP 8 Remove from Σ^h_{ies}, all sets $\Sigma' \in \Sigma^h_{ies}$ for which there exists another set $\Sigma'' \in \Sigma^h_{ies}$ such that $\Sigma' \supseteq \Sigma''$.

paths with embedded indeterminate hidden cycles of G'_d. Using the tree of Fig. 11, we find the following traces: $s_{Y,1} = adccb$, $s_{Y,2} = adcb$, $s_{Y,3} = bdcb$ and $s_{Y,4} = bdccb$. Therefore,

$$S_Y^h = \{adccb,adcb,bdcb,bdccb\}.$$

According to step 1, it is still necessary to obtain the set formed with the projections over $\Sigma_o'^*$ of the traces of S_Y^h, which is given by:

$$P_{oo'}(S_Y^h) = \{s'_1,s'_2,s'_3,s'_4\} = \{acc,ac,c,cc\}.$$

The next step of Algorithm 4 (step 2) requires that sets S_N and S_{NN} be constructed according to Eqs. 18 and 21, respectively and set S_{YY}^{YN} according to Eqs. 20, 25 26 and 27. Using the tree of Fig. 11 and the test automaton G'_{test} of Fig. 10, we obtain:

$$\begin{aligned} S_N &= \{adbb, accc, bccc, bdbb\},\\ S_{NN} &= \emptyset,\\ S_{YY} &= \{adac, acaa, bcaa\}. \end{aligned}$$

Set S_{YY}^{YN} is then formed by taking the longest prefix of each trace of S_{YY} that takes the initial state of G_d to an uncertain state. Using G'_{test} automaton of Fig. 10, it is not hard to check that

$$S_{YY}^{YN} = \{ad,ac,bc\}.$$

We can now move to steps 3 and 4 of Algorithm 4 and form sets $S_{Y,i}^h$ and $S_{N,i}^h$ associated with each $s'_i \in P_{oo'}(S_Y^h)$, $i = 1,2,3,4$. Set $S_{Y,i}^h$, can be easily obtained, being given as:

$$S_{Y,1}^h = \{adccb\},\ S_{Y,2}^h = \{adcb\},\ S_{Y,3}^h = \{bdcb\},\ S_{Y,4}^h = \{bdccb\}.$$

Let us now consider the construction of sets $S_{N,i}^h$, $i = 1,2,3,4$. Consider, initially, $s'_1 = acc$. Since

$$\begin{aligned} S_Y(s'_1) &= \{adcb,bdcb,bdccb\},\\ S_Y^{YN}(s'_1) &= \{ad,bd\},\\ S_N^h(s'_1) &= \{ad,bd,ac,bc,adbb,accc,bccc,bdbb\}, \end{aligned}$$

then $S_{N,1}^h = \{acc\}$. Proceeding in the same manner as above, we obtain $S_N^h(s'_2) = S_N^h(s'_3) = S_N^h(s'_4) = S_N^h(s'_1)$ and thus $S_{N,2}^h = \{ac\}$, $S_{N,3}^h = \{bc\}$ and $S_{N,4}^h = \{bcc\}$. Notice that $S_{N,i}^h$ is formed with the traces of set $S_N^h(s'_i)$ that have as prefixes traces $s_{N,i}$ whose last events belong to Σ'_o and satisfy $P_{oo'}(s_{N,i}) = s'_i$.

Continuing with Algorithm 4, the following sets must be formed in step 5:

$$\begin{aligned} &\Sigma_{Y,1}^1 = \{b,d\},\ \Sigma_{Y,2}^1 = \{b,d\},\ \Sigma_{Y,3}^1 = \{b,d\},\ E_{Y,4}^1 = \{b,d\},\\ &\Sigma_{N,1}^1 = \emptyset,\ \Sigma_{N,2}^1 = \emptyset,\ \Sigma_{N,3}^1 = \{b\},\ \Sigma_{N,4}^1 = \{b\}, \end{aligned}$$

whose events belong to the traces of $S_{Y,i}^h$ and $S_{N,i}^h$ that are not in Σ'_o.

Proceeding according to step 6, we form the following sets:

$$\Sigma^h_{ies,Y1} = \Sigma^h_{ies,Y2} = \Sigma^h_{ies,Y3} = \Sigma^h_{ies,Y4} = \{\{b\},\{d\}\},$$
$$\Sigma^h_{ies,N1} = \Sigma^h_{ies,N2} = \emptyset,\ \Sigma^h_{ies,N3} = \Sigma^h_{ies,N4} = \{\{b\}\}.$$

Still in step 6, the sets obtained above are used to form the following sets:

$$\Sigma^h_{ies,1} = \Sigma^h_{ies,Y1} \cup \Sigma^h_{ies,N1} = \{\{b\},\{d\}\},$$
$$\Sigma^h_{ies,2} = \Sigma^h_{ies,Y2} \cup \Sigma^h_{ies,N2} = \{\{b\},\{d\}\},$$
$$\Sigma^h_{ies,3} = \Sigma^h_{ies,Y3} \cup \Sigma^h_{ies,N3} = \{\{b\},\{d\}\},$$
$$\Sigma^h_{ies,4} = \Sigma^h_{ies,Y4} \cup \Sigma^h_{ies,N4} = \{\{b\},\{d\}\}.$$

Therefore, the innovative event set to deal with the indeterminate hidden cycles of G'_d is, according to step 7, given as:

$$\Sigma^h_{ies} = \Sigma^h_{ies,1} \dot{\times} \Sigma^h_{ies,2} \dot{\times} \Sigma^h_{ies,3} \dot{\times} \Sigma^h_{ies,4} = \{\{b\},\{d\},\{b,d\}\}.$$

Notice that since $\{b,d\} \supset \{b\}$ — $\{b,d\}$ it is also a superset of $\{d\}$ — it must be removed from Σ^h_{ies}, which, therefore, reduces to:

$$\Sigma^h_{ies} = \{\{b\},\{d\}\}.$$

5.4 The search algorithm

In this section, we put together Algorithms 1, 3 and 4 to form Algorithm 5 to carry out the search for minimal diagnosis bases. We assume that for a given G, L is diagnosable with respect to $P_o : \Sigma \to \Sigma_o^*$ and σ_f, where Σ_o is the set of all possible observable events of G. The reader is referred to Table 2 in the Appendix for the notation used in the algorithm.

Remark 10 (Computational complexity of Algorithm 5)
In order to compute all EDESs in the first step of Algorithm 5, we need to build automaton G_d which has $|X_d|$ states and $|X_d| \cdot |\Sigma_o|$ transitions; $|X_d|$ is upper bounded[2] by $2^{|X|}$. If we partition the state space of G_d as $X_d = X^Y_{YN} \dot{\cup} X_Y \dot{\cup} X'_d$, where X'_d accounts for the remaining states, then, using the results of Remark 5, the computational complexity of finding all prime paths starting at all states of X^Y_{YN} using Algorithm 1 is $|X^Y_{YN}| \cdot |\Sigma_o|^{|X_Y|}$ in the worst case. The next computationally intensive step of Algorithm 5 is step 3. At each iteration of that step, it is initially necessary to verify the diagnosability of L with respect to Σ'_o and Σ_f. This could

[2]We point out that practical applications with real systems have yielded diagnosers whose state spaces are of the same order as those of the systems; see *e.g.* Sampath (2001), Sengupta (2001) and Sinnamohideen (2001). In these applications, the worst-case exponential upper bound is far from being attained due to the underlying system structure.

Algorithm 5

STEP 1 Use Algorithm 1, compute Σ_{edes};

STEP 2 Set $\Sigma_{\text{mdbc}} = \Sigma_{\text{edes}}$ and $\Sigma_{\text{mdb}} = \emptyset$;

STEP 3 Pick one of the elements of Σ_{mdbc} with the least cardinality and denote it as Σ'_o, update $\Sigma_{\text{mdbc}} \leftarrow \Sigma_{\text{mdbc}} \setminus \{\Sigma'_o\}$, and compute G'_d;

– If G'_d does not have any indeterminate cycles
  • Set $\Sigma_{\text{mdb}} \leftarrow \Sigma_{\text{mdb}} \cup \{\Sigma'_o\}$;
  [*The algorithm can be stopped here if so desired*]
  • If $\Sigma_{\text{mdbc}} = \emptyset$ then stop. Otherwise, go back to the beginning of step 3.
– Otherwise
  • Use Algorithm 2 to compute all prime paths of G'_{test} and G'_d
  • Use Algorithm 3 to compute Σ^o_{ies};
  • Use Algorithm 4 to compute Σ^h_{ies};
  • Compute $\Sigma_{ies} = \Sigma^o_{ies} \dot{\times} \Sigma^h_{ies}$;
  • Remove from Σ_{ies}, all event sets Σ' for which there exists another set $\Sigma'' \in \Sigma_{ies}$ such that $\Sigma' \supseteq \Sigma''$.
  • Set $\Sigma_{\text{mdbc}} \leftarrow \Sigma_{\text{mdbc}} \cup \left(\{\Sigma'_o\} \dot{\times} \Sigma_{ies}\right)$;
  • Remove from Σ_{mdbc}, all event sets Σ' for which there exists a set $\Sigma'' \in \Sigma_{\text{mdbc}}$ such that $\Sigma' \supseteq \Sigma''$ or there exists a set $\Sigma''' \in \Sigma_{\text{mdb}}$ such that $\Sigma' \supseteq \Sigma'''$.
  • If $\Sigma_{\text{mdbc}} = \{\Sigma_o\}$ then set $\Sigma_{\text{mdb}} = \{\Sigma_o\}$ and stop. Otherwise, go back to the beginning of step 3.

be done in polynomial complexity in the state space of G using verifiers (Moreira et al. 2011); however, since G'_d can be built in linear time in the size of G_d, then diagnosability can be verified in polynomial complexity in the size of G'_d using the indeterminate cycle condition. After that step, we need to compute all prime paths of the currently considered G'_d and G'_{test} using Algorithm 2. Since G'_d has at most $|X_d|$ states and $|X_d| \cdot |\Sigma'_o|$ transitions and $G'_{\text{test}} = G'_d \| G_d$ has at most $|X_d|^2$ states and $|X_d|^2 \cdot |\Sigma_o|$ transitions, the worst-case computational complexity of computing all prime paths of G'_d and G'_{test} is $|\Sigma'_o|^{|X_d|}$ and $|\Sigma_o|^{|X_d|^2}$, respectively (using again the results of Remark 5). The subsequent calculations in step 3, including the application of Algorithms 3 and 4, involve manipulations of sets of innovative events built from the prime paths of G'_d and G'_{test}; these calculations have worst-case complexity of $2^{|\Sigma_{ies}|}$, which is below that of the preceding calculation. After the stopping condition, step 3 can be further iterated if the goal is to compute more than one (or all) minimal diagnosis bases. The total number of iterations of step 3, for computing all diagnosis bases, is upper bounded by $2^{|\Sigma_o \setminus \Sigma_{\text{edes,min}}|}$ where $\Sigma_{\text{edes,min}}$ is the set with minimum cardinality among the elements of Σ_{edes}.

We now illustrate the search algorithm above by means of two examples.

Example 5 Let us consider the state transition diagram of automaton G depicted in Fig. 6. As shown in Example 3, L is diagnosable with respect to $P_o : \Sigma^* \to \Sigma_o^*$ and

Σ_f, where $\Sigma_o = \{a,b,c,d\}$ and $\Sigma_f = \{\sigma_f\}$. Therefore, Algorithm 5 can be applied to search for all minimal diagnosis bases of L.

The first step of Algorithm 5 is the computation of the set of elementary diagnosing event sets Σ_{edes}. This set has already been obtained in Example 3, being given as $\Sigma_{\text{edes}} = \{\{a,c\}\}$. Therefore, according to step 2 of Algorithm 5, $\Sigma_{\text{mdbc}} = \{\{a,c\}\}$. As we can see, there is a unique minimal diagnosis basis candidate, and thus $\Sigma'_o = \{a,c\}$. As shown in Example 3, L is not diagnosable with respect to $P'_o : \Sigma^* \to \Sigma'^*_o$ and Σ_f. As a consequence, it is necessary to compute the innovative event set Σ_{ies} in order to enlarge Σ'_o. This problem has already been addressed in Examples 3 and 4, leading to $\Sigma^o_{ies} = \{\{d\}\}$ and $\Sigma^h_{ies} = \{\{b\},\{d\}\}$. Therefore,

$$\Sigma_{ies} = \Sigma^o_{ies} \dot{\times} \Sigma^h_{ies} = \{\{b,d\},\{d\}\}.$$

Since $\{d\} \subset \{b,d\}$, then $\{b,d\}$ must be removed from Σ_{ies}, and thus the next minimal diagnosis basis candidate set is given by:

$$\Sigma_{\text{mdbc}} = \Sigma_{\text{mdbc}} \cup \left(\{\Sigma'_o\} \dot{\times} \Sigma_{ies}\right) = \emptyset \cup \{\{a,c,d\}\} = \{\{a,c,d\}\}.$$

We should now run step 3 again with $\Sigma'_o = \{a,c,d\}$. It can be checked that the corresponding partial diagnoser G'_d has no indeterminate cycles. We may, therefore, conclude that L is diagnosable with respect to P'_o and Σ_f, which implies that

$$\Sigma_{\text{mdb}} = \{\Sigma'_o\} = \{\{a,c,d\}\}.$$

Since $\Sigma_{\text{mdbc}} = \emptyset$, the algorithm must stop, and thus $\{a,c,d\}$ is the unique minimal diagnosis basis for L.

Example 6 Let us consider the problem of computing all minimal diagnosis bases for the language L generated by automaton G whose state transition diagram is depicted in Fig. 12a. The set of events of G is $\Sigma = \{a,b,c,d,e,g,\sigma,\sigma_f\}$, and its sets of observable, unobservable and fault events are, respectively, $\Sigma_o = \{a,b,c,d,e,g\}$ and $\Sigma_{uo} = \{\sigma,\sigma_f\}$ and $\Sigma_f = \{\sigma_f\}$. The corresponding diagnoser is shown in Fig. 12b, where it is clear that G_d has no indeterminate cycles. Thus, we may conclude that L is diagnosable with respect to P_o and Σ_f. We can therefore use Algorithm 5 to find all minimal diagnosis bases for L.

According to Algorithm 5, the first step is to find all elementary diagnosing event sets of G_d. Following Algorithm 1, we obtain $\Sigma_{\text{edes}} = \{\{b,d\},\{d,e\}\}$. Moving to step 2 of Algorithm 5, we must set $\Sigma_{\text{mdbc}} = \Sigma_{\text{edes}}$ and $\Sigma_{\text{mdb}} = \emptyset$.

In order to continue with the algorithm, we must choose, among the sets in Σ_{mdbc}, the one with the smallest cardinality. However all sets in Σ_{mdbc} have the same cardinality, and thus any set can be chosen arbitrarily. Choosing $\Sigma'_o = \{b,d\}$ then, according to step 3, the set of minimal diagnosis basis candidates becomes $\Sigma_{\text{mdbc}} = \{\{d,e\}\}$. The partial diagnoser G'_d for $\Sigma'_o = \{b,d\}$ is depicted in Fig. 13, and since it has one indeterminate hidden cycle, L is not diagnosable with respect to $P_o{}'$.

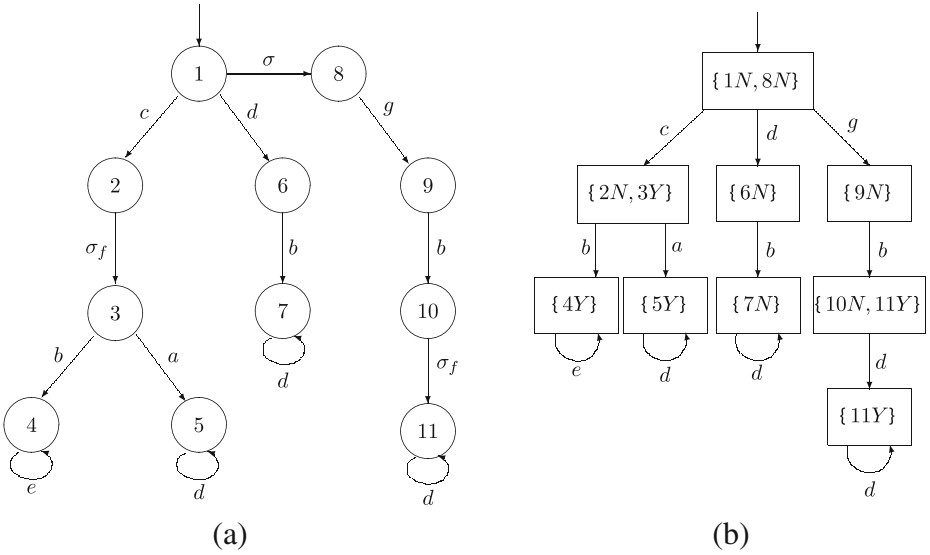


Fig. 12 Automaton G and the corresponding centralized diagnoser for Example 6

In general, Algorithms 3 and 4 must be used to compute Σ^o_{ies} and Σ^h_{ies}, respectively. However, since G'_d has indeterminate hidden cycles only, there is no need to go through all the steps of Algorithm 3, *i.e.*, $\Sigma^o_{ies} = \{\emptyset\}$. Moving to Algorithm 4, we need to build automaton G'_{test} depicted in Fig. 14, and, based on G'_{test}, we construct the tree of Fig. 15. Following the first step of Algorithm 4, we can see that $S^h_Y = \{cbe\}$ and $P_{oo'}(S^h_Y) = \{b\}$. Therefore $s'_1 = b$ is the unique sequence of L'_d to be considered, which implies that $S^h_Y(s'_1) = \{cbe\}$ and $S^h_N(s'_1) = \{c,gb,dbd\}$ since $S^{YN}_Y(s'_1) = \emptyset$, $S^{YN}_{YY} = \{c,gb\}$, $S_N = \emptyset$ and $S_{NN} = \{dbd\}$. Therefore, $S^h_{Y,1} = \{cbe\}$ and $S^h_{N,1} = \{gb\}$ and so $\Sigma_{ies} = \Sigma^o_{ies} \dot{\times} \Sigma^h_{ies} = \{\{c\},\{e\},\{g\}\}$. At the end of step 3, we obtain $\Sigma_{\text{mdbc}} = \{\{d,e\},\{b,c,d\},\{b,d,g\}\}$, and since $\Sigma_{\text{mdbc}} \neq \{\Sigma_o\}$, we must go back to the beginning of step 3.

In the second run of step 3, we choose $\Sigma'_o = \{d,e\}$ since it is the smallest cardinality set in Σ_{mdbc}, and thus $\Sigma_{\text{mdbc}} = \{\{b,c,d\},\{b,d,g\}\}$. The computation of G'_d

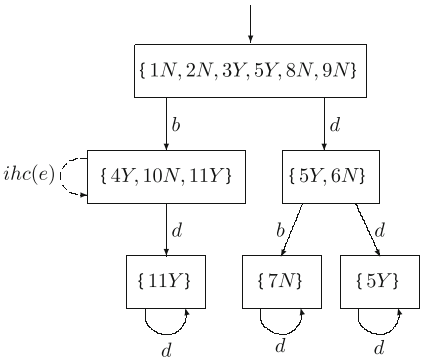


Fig. 13 Partial diagnoser G'_d assuming $\Sigma'_o = \{b,d\}$ for Example 6

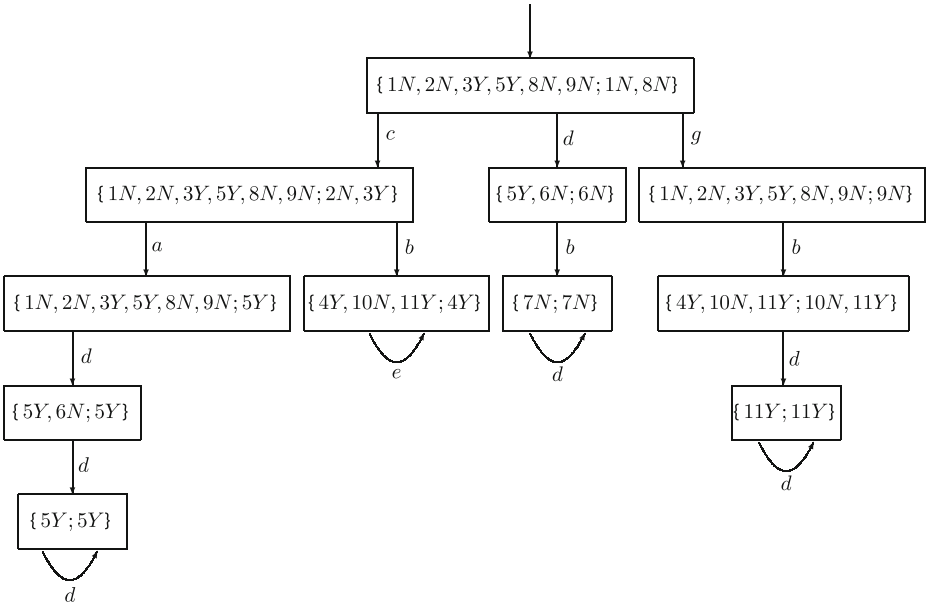


Fig. 14 Test automaton G'_{test} assuming $\Sigma'_o = \{b, d\}$ for Example 6

(not depicted in the paper) shows that Σ'_o is not a minimal diagnosis basis for L, since G'_d has an indeterminate observed cycle. Therefore $\Sigma^h_{ies} = \{\emptyset\}$ and following the steps of Algorithm 3, we obtain $\Sigma^o_{ies} = \{\{b\}, \{a, g\}, \{c, g\}\}$. It is not difficult to

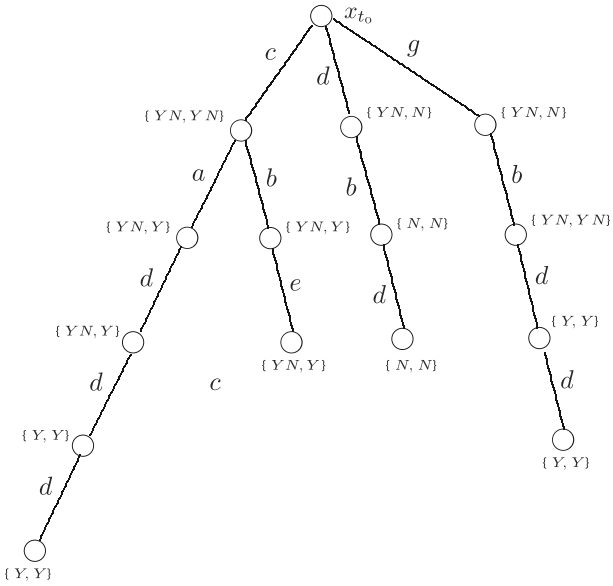


Fig. 15 Tree corresponding to test automaton G'_{test} assuming $\Sigma'_o = \{b, d\}$ for Example 6

check that $\Sigma_{ies} = \Sigma^{o}_{ies}$. At the end of the second run of step 3, we have $\Sigma_{\text{mdbc}} = \{\{b,d,e\},\{b,c,d\},\{b,d,g\},\{a,d,e,g\},\{c,d,e,g\}\}$.

Running step 3 five more times to teste each element of Σ_{mdbc}, we can see that, all elements of Σ_{mdbc} are diagnosis basis. Therefore,

$$\Sigma_{\text{mdb}} = \{\{b,d,e\},\{b,c,d\},\{b,d,g\},\{a,d,e,g\},\{c,d,e,g\}\}$$

is the set of all minimal diagnosis bases for L.

6 Conclusion

We have investigated the construction of minimal diagnosis bases by exploiting structural properties of diagnoser automata. This approach is different from prior work, which primarily focused on enumerative search methods over the power set of the set of potentially observable events. By constructing what we termed partial diagnosers and test diagnosers, and examining where violations of diagnosability occur in their transition structures, we have discovered rules for guiding the update of the set of observable events and used them to develop algorithmic procedures that search for diagnosis bases and identify their minimal elements. In some sense, this approach is reminiscent of the strategy of counter-example guided search often used in combinatorial optimization. Our overall procedure is embodied into Algorithm 5, which takes as input the given automaton and outputs all (if so desired) minimal diagnosis bases and their corresponding diagnosers. We are currently investigating the adaptation of the techniques in this paper to verifier automata of the type recently introduced in Moreira et al. (2011), in place of diagnoser automata.

Acknowledgements We would like to thank the anonymous reviewers for their comments and suggestions which helped improve the presentation and readability of the paper. The research work of João Carlos Basilio has been supported by the Brazilian Research Council (CNPq), grants 200820/2006-0 and 307939/2007-3. The research of Stéphane Lafortune has been supported in part by NSF grants ECCS-0624821 and CNS-0930081.

Appendix

We present here two tables: Table 1 lists all the acronyms and Table 2 presents the main notation used in the paper.

Table 1 List of acronyms

Acronym	Name
EDES	Elementary diagnosing even set
FPES	Faulty path event set
FPOSS	Faulty path origin state set

Table 2 Notation

Notation	Meaning
$P_o : \Sigma^* \to \Sigma_o^*$	Natural projection from Σ^* to Σ_o^*
$P'_o : \Sigma^* \to \Sigma_o'^*$	Natural projection from Σ^* to $\Sigma_o'^*$
$P''_o : \Sigma^* \to \Sigma_o''^*$	Natural projection from Σ^* to $\Sigma_o''^*$
$P_{oo'} : \Sigma_o^* \to \Sigma_o'^*$	Natural projection from Σ_o^* to $\Sigma_o'^*$
$P_{oo''} : \Sigma_o^* \to \Sigma_o''^*$	Natural projection from Σ_o^* to $\Sigma_o''^*$
Σ_{edes}	Set of elementary diagnosing event sets
Σ_{mdb}	Minimal diagnosis basis set
Σ_{mdbc}	Minimal diagnosis basis candidate set
Σ_{ies}^{o}	Set of innovative event sets associated with the indeterminate observed cycles in G'_d
Σ_{ies}^{h}	Set of innovative event sets associated with the indeterminate hidden cycles in G'_d
$G'_{\text{test}} = G'_d \parallel G_d$	Test automaton to verify language diagnosability of G'_d
L, L_d, L'_d, L'_{test}	Languages generated by automata G, G_d, G'_d, and G'_{test}, respectively
X_{YN}^{Y}	Set of all uncertain states of G_d from which there exists a transition that takes to a certain state
$X_Y X_N$	Sets of all certain and normal states of G_d, respectively
S'_t	Set of the traces associated with all prime paths of G'_{test}
S_{YY}	Set of all traces $s \in S'_t$ such that there exists a leaf labeled as (x'_d, x_d) where x'_d and x_d are both normal and $f_t(x_{t_0}, s) = (x'_d, x_d)$
S_{NN}	Set of all traces $s \in S'_t$ such that there exists a leaf labeled as (x'_d, x_d) where x'_d and x_d are both normal and $f_t(x_{t_0}, s) = (x'_d, x_d)$
S_Y	Set formed with all Y-prime paths of G'_{test}
S_N	Set of all traces associated with prime paths of G'_{test} whose first components of the states of the unique cyclic path are uncertain states of an indeterminate cycle (observed or hidden) of G'_d and the second components are either normal states of G_d that are not states of an indeterminate cycle in G'_d

References

Basilio JC, Lafortune S (2009) Robust codiagnosability of discrete event systems. In: Proceedings of the American control conference, pp 2202–2209

Boel RK, van Schuppen JH (2002) Decentralized failure diagnosis with costly communication between diagnosers. In: Proceedings of the 6th international workshop on discrete event systems, pp 175–181

Cabasino MP, Giua A, Seatzu C (2010) Fault detection for discrete event systems using petri nets with unobservable transitions. Automatica 46(9):1531–1539

Cassandras CG, Lafortune S (2008) Introduction to discrete event systems, 2nd edn. Springer, Boston

Cassez F, Tripakis S (2008) Fault diagnosis with static and dynamic observers. Fundam Inform 88(4):497–540

Dallal E, Lafortune S (2010) On most permissive observers in dynamic sensor optimization problems for discrete event systems. In: Proceedings of the 48th annual allerton conference on communication, control, and computing, pp 318–324

Debouk R, Lafortune S, Teneketzis D (2002) On an optimization problem in sensor selection. Discrete Event Dynamic Systems: Theory and Applications 12(4):417–445

Fabre E, Benveniste A, Haar S, Jard C (2005) Distributed monitoring of concurrent and asynchronous systems. Discrete Event Dyn Syst: Theory Appl 15(1):33–84

Garcia HE, Yoo TS (2005) Model-based detection of routing events in discrete flow networks. Automatica 41(4):583–594

Genc S (2008) Formal methods for intrusion detection of windows nt attacks. In: 3rd annual symposium on information assurance (ASIA '08) & 11th annual NYS cyber security conference, vol 1, pp 71–79

Haar S (2010) What topology tells us about diagnosability in partial order semantics. In: Proceedings of the 10th international workshop on discrete event systems, pp 221–226

Itai A, Lipton RJ, Papadimitriou CH, Rodeh M (1981) Covering graphs by simple circuits. SIAM J Comput 10(4):746–750

Jéron T, Marchand H, Genc S, Lafortune S (2008) Predictability of sequence patterns in discrete event systems. In: Proceedings of the 17th IFAC world congress, pp 537–543

Jiang S, Huang Z, Chandra V, Kumar R (2001) A polynomial algorithm for testing diagnosability of discrete-event systems. IEEE Trans Automat Contr 46(8):1318–1321

Jiang S, Kumar R, Garcia H (2003) Optimal sensor selection for discrete-event systems with partial observation. IEEE Trans Automat Contr 48(3):369–381

Jiang SB, Kumar R (2004) Failure diagnosis of discrete-event systems with linear-time temporal logic specifications. IEEE Trans Automat Contr 49(6):934–945

Johnson DB (1975) Finding all the elementary circuits of a directed graph. SIAM J Comput 4(1): 77–84

Kumar R, Takai S (2009) Inference-based ambiguity management in decentralized decision-making: decentralized diagnosis of discrete-event systems. IEEE Trans Autom Sci Eng 6(3):479–491

Lafortune S, Teneketzis D, Sampath M, Sengupta R, Sinnamohideen K (2001) Failure diagnosis of dynamic systems: an approach based on discrete event systems. In: Proceedings of the American control conference, vol 3, pp 2058–2071

Lin F (1994) Diagnosability of discrete-event systems and its applications. Discrete Event Dyn Syst: Theory Appl 4(2):197–212

Lunze J, Schroder J (2004) Sensor and actuator fault diagnosis of systems with discrete inputs and outputs. IEEE Trans Syst Man Cybern, Part B, Cybern 34(2):1096–1107

Moreira MV, Jesus TC, Basilio JC (2011) Polynomial time verification of decentralized diagnosability of discrete event systems. IEEE Trans Automat Contr 56(7):1679–1684

Pandalai DN, Holloway LE (2000) Template languages for fault monitoring of timed discrete event processes. IEEE Trans Automat Contr 45(5):868–882

Pencolé Y, Cordier MO (2005) A formal framework for the decentralized diagnosis of large scale discrete event systems and its applications to telecommunication networks. Artif Intell 164 (1–2):121–170

Ramadge PJ, Wonham WM (1989) The control of discrete-event systems. Proc IEEE 77(1): 81–98

Sampath M (2001) A hybrid approach to failure diagnosis of industrial systems. In: Proceedings of the American control conference, vol 3, pp 2077–2082

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1995) Diagnosability of discrete-event systems. IEEE Trans Automat Contr 40(9):1555–1575

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1996) Failure diagnosis using discrete event models. IEEE Trans Control Syst Technol 4(2):105–124

Sengupta R (2001) A discrete event approach for vehicle failure diagnostics. In: Proceedings of the American control conference, vol 3, pp 2083–2086

Sinnamohideen K (2001) Discrete-event diagnostics of heating, ventilation, and air-conditioning systems. In: Proceedings of the American control conference, vol 3, pp 2072–2076

Thorsley D, Teneketzis D (2005) Diagnosability of stochastic discrete-event systems. IEEE Trans Automat Contr 50(4):476–492

Thorsley D, Teneketzis D (2007) Active acquisition of information for diagnosis and supervisory control of discrete event systems. Discrete Event Dyn Syst: Theory Appl 17(4):531–583

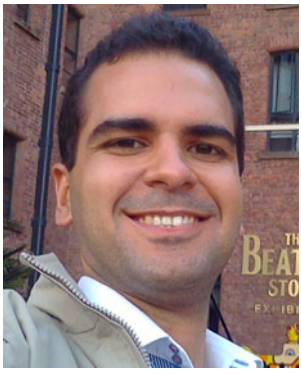
Tripakis S (2002) Fault diagnosis for timed automata. In: Formal techniques in real time and fault tolerant systems (FTRTFT). Lecture notes in computer sciences, vol 2469, pp 205–222. Springer-Verlag, New York

Wang W, Lafortune S, Girard AR, Lin F (2010) Optimal sensor activation for diagnosing discrete event systems. Automatica 46(7):1165–1175

Wang Y, Yoo TS, Lafortune S (2007) Diagnosis of discrete event systems using decentralized architectures. Discrete Event Dyn Syst: Theory Appl 17(2):233–263

Yoo TS, Lafortune S (2002) NP-completeness of sensor selection problems arising in partially observed discrete-event systems. IEEE Trans Automat Contr 47(9):1495–1499
Yoo TS, Lafortune S (2002) Polynomial-time verification of diagnosability of partially observed discrete-event systems. IEEE Trans Automat Contr 47(9):1491–1495

João Carlos Basilio was born on March 15, 1962 in Juiz de Fora, Brazil. He received the Electrical Engineering degree in 1986 from the Federal University of Juiz de Fora, Juiz de Fora, Brazil, the M.Sc. degree in Control from the Military Institute of Engineering, Rio de Janeiro, Brazil, in 1989, and the Ph.D. degree in Control from Oxford University, Oxford, U.K., in 1995. He began his career in 1990 as an Assistant Lecturer at the Department of Electrical Engineering of the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, and, since 2007, has been a Senior Associate Professor in Control at the same department. He served as the Academic Chair for the graduation course in Control and Automation from January, 2005, to December, 2006, and as the Chair for the Electrical Engineering Post-graduation Program from January, 2008, to February, 2009. From September, 2007, to December, 2008, he spent a sabbatical leave at the University of Michigan, Ann Arbor. His is currently interested in discrete-event systems and in the development of control and automation laboratories and new teaching techniques. Dr. Basilio is the recipient of the Correia Lima Medal.



Saulo Telles Souza Lima was born in Rio de Janeiro, Brazil, in 1985. He received the Electrical Engineering degree and the M.Sc. degree in Control from the Federal University of Rio de Janeiro, in

2009 and 2010, respectively. Since 2010, he has been working at the Brazilian Oil Company Petrobras in the development of undersea electrical systems for oil processing and boosting in deep and ultra deep water depth.

Stéphane Lafortune received the B. Eng. degree from Ecole Polytechnique de Montréal in 1980, the M. Eng. degree from McGill University in 1982, and the Ph.D. degree from the University of California at Berkeley in 1986, all in electrical engineering. Since September 1986, he has been with the University of Michigan, Ann Arbor, where he is a Professor of Electrical Engineering and Computer Science. Dr. Lafortune is a Fellow of the IEEE (1999). He received the Presidential Young Investigator Award from the National Science Foundation in 1990 and the George S. Axelby Outstanding Paper Award from the Control Systems Society of the IEEE in 1994 (for a paper co-authored with S. L. Chung and F. Lin) and in 2001 (for a paper co-authored with G. Barrett). Dr. Lafortune's research interests are in discrete event systems and include multiple problem domains: modeling, diagnosis, control, optimization, and applications to computer systems. He is the lead developer of the software package UMDES and co-developer of DESUMA with L. Ricker. He co-authored, with C. Cassandras, the textbook *Introduction to Discrete Event Systems—Second Edition* (Springer, 2008). Dr. Lafortune is a member of the editorial boards of the Journal of Discrete Event Dynamic Systems: Theory and Applications and of the International Journal of Control.

Marcos Vicente Moreira was born on May, 11, 1976 in Rio de Janeiro, Brazil. He received the Electrical Engineer degree, the M.Sc. degree and the D. Sc. degree in Control from the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, in 2000, 2002 and 2006, respectively. Since 2007, he has been an Associate Professor at the Department of Electrical Engineering at the Federal University of Rio de Janeiro. His main interests are multivariable control, robust control, discrete-event systems and the development of control laboratory techniques.