Temporal Diagnosis of Discrete-Event Systems with Dual Knowledge Compilation

Nicola Bertoglio, Gianfranco Lamperti, Marina Zanella

Temporal Diagnosis of Discrete-Event Systems with Dual Knowledge Compilation*

Nicola Bertoglio[0000−0002−7905−5957], Gianfranco Lamperti[0000−0002−1915−6932], and Marina Zanella[0000−0003−3896−3913]

Department of Information Engineering, University of Brescia, Brescia, Italy
{n.bertoglio001,gianfranco.lamperti,marina.zanella}@unibs.it

Abstract. Diagnosis aims to explain the abnormal behavior of a system based on the symptoms observed. In a discrete-event system (DES), the symptom is a temporal sequence of observations. At the occurrence of each observation, the diagnosis engine generates a set of candidates, a candidate being a set of faults: such a process requires costly model-based reasoning. This is why a variety of knowledge compilation techniques have been proposed; the most notable of them relies on a *diagnoser* and requires both the diagnosability of the DES and the generation of the whole system space. To avoid both diagnosability and total knowledge compilation, while preserving efficiency, a diagnosis technique is proposed, which is inspired by the two operational modes of the human mind. If the symptom of the DES is part of the knowledge or experience of the diagnosis engine, then *Engine 1* allows for efficient diagnosis. If, instead, the symptom is unknown, then *Engine 2* comes into play, which is far less efficient than *Engine 1*. Still, the experience acquired by *Engine 2* is then integrated into the *temporal dictionary* of the DES, which allows for diagnosis in linear time. This way, if the same problem arises anew, then it will be solved by *Engine 1* efficiently. The temporal dictionary can also be extended by specialized knowledge coming from *scenarios*, which are behavioral patterns of the DES that need to be diagnosed quickly. As such, the temporal dictionary is *open* and relies on *dual knowledge compilation*.

Keywords: Diagnosis · Discrete-Event Systems · Automata · Temporal Dictionary · Scenarios · Temporal Explanation · Preprocessing · Knowledge Compilation · Symptom Patterns · Abduction.

1 Introduction

Diagnosis aims at explaining the abnormal behavior of a system based on the observations relevant to its operation that are perceived from the outside. In the Artificial Intelligence community, the definition of the task [23] led to the model-based paradigm [6], according to which the normal behavior of the system to be

* This work was supported in part by Lombardy Region (Italy), project *Smart4CPPS, Linea Accordi per Ricerca, Sviluppo e Innovazione, POR-FESR 2014-2020 Asse I.*

diagnosed is described by a model and the diagnosis results have to explain the discrepancies between what has been observed at the system output terminals and what we expected to observe on grounds of the model itself. The diagnosis task produces a collection of sets of faulty components, where each set, called a *candidate*, is an explanation of the observation. Each candidate explains the observation as assuming that all the components in the candidate are not behaving normally and all the others are behaving normally is consistent with the observation. This *consistency-based* diagnosis was initially conceived for static systems, such as combinational circuits. For a dynamical system, a discrete-event system (DES) [3] model can be adopted, this being a finite automaton. This model is typically distributed, consisting of several automata that communicate with one another [2]. Although consistency-based diagnosis is applicable to DESs by modeling their nominal behavior only [22], a DES specification usually involves its abnormal behavior also, as in the seminal work by Sampath et al. [25]. The input of the diagnosis task for a DES is a temporal sequence of observations; the output is a set of candidates, each candidate being a set of faults, where a fault is associated with an abnormal state transition represented in the DES model. Diagnosing a DES becomes a form of *abductive* reasoning, inasmuch the candidates are generated based on the trajectories (sequences of state transitions) of the DES that entail the sequence of observations. The approach in [25] relies on a *diagnoser*, a data structure that is derived in a preprocessing phase from the *space* (or global model) of the DES. Such a diagnoser is exploited on line, in order to generate a new set of candidate diagnoses upon perceiving each observation. However, this method requires the generation of the global model of the DES, which is impractical even for distributed systems of moderate size, owing to a combinatorial state explosion. Moreover, in order for the diagnoser to produce a sound and complete set of candidates, the DES is required to fulfill a formal property called *diagnosability*. By definition, a DES is *diagnosable* if every fault occurred can be detected within a finite number of observable transitions of the DES while it is moving in a trajectory of its space. Unsurprisingly, the problem of checking diagnosability has given rise to an extended literature in the last two decades [5, 8, 30, 4, 21, 28, 24, 20, 26, 19, 29, 27]. One alternative to the diagnoser approach is the *active-system* approach [1, 11–13, 15], which neither requires the generation of the global model nor the diagnosability of the DES. The rationale behind the traditional active-system approach is to perform the abduction online, a possibly costly operation that, however, being driven by the sequence of observations, can only focus on the trajectories that produce such a sequence. This paper, which stems from the active-system approach, proposes a novel, more efficient, method to compute a new set of candidate diagnoses of a DES upon receiving each observation. The candidates generated by this technique are endowed with a property, called *temporal explanation*, which has so far been missing in the active-system approach. Temporal explanation is supported by a technique, embedded in the diagnosis engine, called *backward pruning*. Efficiency is achieved by preprocessing the system model to construct a data structure, called a *temporal dictionary*, which is exploited online, when the

DES is being operated. In contrast with the diagnoser, the temporal dictionary is not the result of total knowledge compilation, instead it embodies the knowledge relevant to some selected (domain-dependent) behaviors, called *scenarios*. In addition, whenever a sequence of observations that is not encompassed by the dictionary is processed, the dictionary is extended online. In other words, the dictionary is *open* and relies on *dual knowledge compilation*.

2 The Two-Systems Metaphor of the Mind

According to Daniel Kahneman [9], psychologist and Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2002, two modes of thinking coexist in the human brain, which correspond to two systems in the mind, called *System 1* and *System 2*. *System 1* operates automatically and quickly, with little if any effort, and no sense of voluntary control, such as when orienting to the source of a sudden sound or driving a car in an empty road. By contrast, *System 2* operates consciously and slowly, with attention being focused on demanding mental activities, possibly including complex computations or inferences, such as when filling out an intricate application form or checking the validity of a complex argument. Intriguingly, an activity initially performed by *System 2*, such as driving a car or playing the piano, may be subsequently operated by *System 1* after appropriate training. This "dual system" architecture of the mind is a metaphor for the diagnosis approach for DESs presented in this paper. The proposed diagnosis engine (DE) operates in two different modes resembling *System 1* and *System 2* in the human mind, called *Engine 1* and *Engine 2*. If the diagnosis problem to be solved is part of the knowledge or experience of the DE, then *Engine 1* can solve this problem quickly. If, instead, the problem is not part of the knowledge or experience of the DE, then comes into play *Engine 2*, which requires deep model-based reasoning and, therefore, operates far more slowly than *Engine 1*. Still, the experience acquired by *Engine 2* in solving the diagnosis problem can be integrated into the knowledge of the DE, so that, in the future, the same diagnosis problem can be solved by *Engine 1* efficiently. Besides, the DE is not born naked, that is, without any knowledge except the model of the DES, otherwise *Engine 2* would operate far more frequently than *Engine 1* for a possibly long time. Instead, the DE starts working being already equipped with specialized knowledge on domain-dependent *scenarios* that are considered either most probable or most critical for the safety of the DES (or the surrounding environment) and, as such, need to be coped with efficiently.

3 Discrete-Event Systems

A DES is assumed to be a network of *components*, where each component is endowed with input and output *pins* and is modeled as a communicating automaton [2]. Each output pin of a component is connected with an input pin of another component by a *link*. The mode in which a transition is triggered in a component is threefold: (1) spontaneously (formally, by the empty event ε),

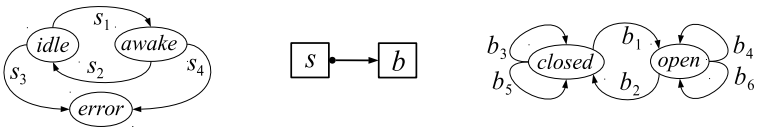


Fig. 1. DES $\mathcal{P}$ (center) and models of the sensor s (left) and the breaker b (right).

Component transition	*Description*
$s_1 = \langle idle, (ko, \{op\}), awake\rangle$	s detects a threatening event and commands b to open
$s_2 = \langle awake, (ok, \{cl\}), idle\rangle$	s detects a liberating event and commands b to close
$s_3 = \langle idle, (ko, \{cl\}, error\rangle$	s detects a threatening event, yet commands b to close
$s_4 = \langle awake, (ok, \{op\}), error\rangle$	s detects a liberating event, yet commands b to open
$b_1 = \langle closed, (op, \emptyset), open\rangle$	b reacts to the opening command by opening
$b_2 = \langle open, (cl, \emptyset), closed\rangle$	b reacts to the closing command by closing
$b_3 = \langle closed, (op, \emptyset), closed\rangle$	b does not react to the opening command
$b_4 = \langle open, (cl, \emptyset), open\rangle$	b does not react to the closing command
$b_5 = \langle closed, (cl, \emptyset), closed\rangle$	b reacts to the closing command by remaining closed
$b_6 = \langle open, (op, \emptyset), open\rangle$	b reacts to the opening command by remaining open

Table 1. Transition details for sensor (top) and breaker (bottom) in the DES $\mathcal{P}$.

(2) by an (external) event coming from the extern of the DES, or (3) by an (internal) event coming from another component of the DES. When a component performs a transition, it possibly generates new events on its output pins, which possibly trigger the transitions of other components, where the triggering events are consumed. A transition generating an output event on a link can occur only if this link is not occupied by another event already.

Example 1. Centered in Figure 1 is a DES called $\mathcal{P}$ (protection) which includes two components, a sensor s and a breaker b, and one link connecting the (single) output pin of s with the (single) input pin of b. The model of s (outlined on the left-hand side) involves three states (denoted by circles) and four transitions (denoted by arcs). The model of b (outlined on the right-hand side) involves two states and six transitions. Each component transition t from a state p to a state p', triggered by an input event e, and generating a set of output events E, is denoted by the (angled) triple $t = \langle p, (e, E), p'\rangle$, as detailed in Table 1.

For diagnosis purposes, we need to characterize a DES $\mathcal{X}$ with its *observability* (whether each transition is observable or unobservable) and *normality* (whether each transition is normal or faulty). To this end, let $\mathbf{T}$ be the set of component transitions in $\mathcal{X}$, $\mathbf{O}$ a finite set of *observations*, and $\mathbf{F}$ a finite set of *faults*. The *mapping table* of $\mathcal{X}$ is a function $\mu(\mathcal{X}) : \mathbf{T} \mapsto (\mathbf{O} \cup \{\varepsilon\}) \times (\mathbf{F} \cup \{\varepsilon\})$, where ε is the *empty* symbol. The table $\mu(\mathcal{X})$ can be represented as a finite set of triples (t, o, f), where $t \in \mathbf{T}$, $o \in \mathbf{O} \cup \{\varepsilon\}$, and $f \in \mathbf{F} \cup \{\varepsilon\}$. The triple (t, o, f) defines the observability and normality of t: if $o \neq \varepsilon$, then t is *observable*, else t is *unobservable*; if $f \neq \varepsilon$, then t is *faulty*, else t is *normal*.

Example 2. With reference to the DES $\mathcal{P}$ introduced in Example 1, the mapping table $\mu(\mathcal{P})$ includes the following triples: (s_1, act, ε), (s_2, sby, ε), (s_3, act, fos), (s_4, sby, fcs), (b_1, opn, ε), (b_2, cls, ε), (b_3, ε, fob), (b_4, ε, fcb), $(b_5, \varepsilon, \varepsilon)$, and $(b_6, \varepsilon, \varepsilon)$, where the symbols have the following meaning: act = activate, sby = standby, opn = open, cls = closed, fos = failed to command to open, fcs = failed to command to close, fob = failed to open, fcb = failed to close.

At each time instant, a DES $\mathcal{X}$ is in a state $x = (C, L, \delta)$, where C is the array of the current states of the components, L is the array of the (possibly empty) events currently placed on the links, and δ is the (possibly empty) set of faults occurred in $\mathcal{X}$ starting from its initial state $x_0 = (C_0, L_0, \emptyset)$. The occurrence of a component transition t moves $\mathcal{X}$ from a state x to a state x', in other words, a transition $\langle x, t, x' \rangle$ occurs in $\mathcal{X}$. Hence, assuming that only one component transition at a time can occur, the process that moves a DES from its initial state to another state is represented by a sequence of component transitions, called a *trajectory* of the DES. The set of possible trajectories of $\mathcal{X}$ is specified by a deterministic finite automaton (DFA) called the *diagnosis space* of $\mathcal{X}$, namely,

$$\mathcal{X}^* = (\Sigma, X, \tau, x_0) \tag{1}$$

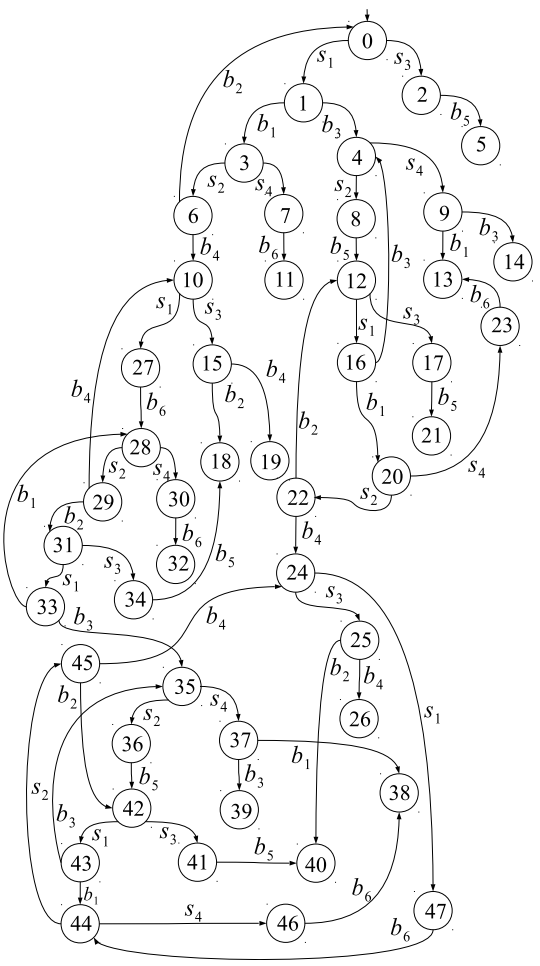
where Σ (the alphabet) is the set of component transitions, X is the set of states, τ is the (deterministic) transition function, $\tau : X \times \Sigma \mapsto X$, and x_0 is the initial state.[1] Thus, each string $T = [t_1, \ldots, t_n]$ of the regular language of $\mathcal{X}^*$ is a trajectory of $\mathcal{X}$. Based on the mapping table $\mu(\mathcal{X})$, each trajectory $T \in \mathcal{X}^*$ is associated with one *symptom* and one *diagnosis*. The *symptom* $\mathcal{O}$ of T is the finite *sequence* of observations involved in T,

$$\mathcal{O} = [\, o \mid t \in T, (t, o, f) \in \mu(\mathcal{X}), o \neq \varepsilon \,]. \tag{2}$$

The *diagnosis* δ of T is the set of faults marking the accepting state of T in $\mathcal{X}^*$. Since a diagnosis is a set, at most one instance of each fault f can be in δ. Hence, generally speaking, the domain of possible diagnoses is the powerset $2^{\mathbf{F}}$, which is finite. By contrast, several instances of the same observation can be in the symptom $\mathcal{O}$; therefore, the domain of possible symptoms is in general infinite. We say that the trajectory T *implies* both $\mathcal{O}$ and δ, denoted $T \Rightarrow \mathcal{O}$ and $T \Rightarrow \delta$, respectively. Since a trajectory of $\mathcal{X}$ is observed as a symptom and since the observed symptom can be implied by several (possibly infinite) trajectories, it follows that several diagnoses can be associated with the same symptom, which are collectively called the *explanation* of the symptom. Formally, let $\mathcal{O}$ be a symptom of $\mathcal{X}$ and $\delta(T)$ denote the diagnosis of T. The explanation Δ of $\mathcal{O}$ is the finite set of diagnoses, called *candidates*, defined as

$$\Delta(\mathcal{O}) = \{\, \delta(T) \mid T \in \mathcal{X}^*, T \Rightarrow \mathcal{O} \,\}. \tag{3}$$

[1] Implicitly, all states of $\mathcal{X}^*$ are also accepting (final) states.



p^*	$C = (s, b)$	L	δ
0	$(idle, closed)$	ε	$\emptyset$
1	$(awake, closed)$	op	$\emptyset$
2	$(error, closed)$	cl	$\{fos\}$
3	$(awake, open)$	ε	$\emptyset$
4	$(awake, closed)$	ε	$\{fob\}$
5	$(error, closed)$	ε	$\{fos\}$
6	$(idle, open)$	cl	$\emptyset$
7	$(error, open)$	op	$\{fcs\}$
8	$(idle, closed)$	cl	$\{fob\}$
9	$(error, closed)$	op	$\{fob, fcs\}$
10	$(idle, open)$	ε	$\{fcb\}$
11	$(error, open)$	ε	$\{fcs\}$
12	$(idle, closed)$	ε	$\{fob\}$
13	$(error, open)$	ε	$\{fob, fcs\}$
14	$(error, closed)$	ε	$\{fob, fcs\}$
15	$(error, open)$	cl	$\{fcb, fos\}$
16	$(awake, closed)$	op	$\{fob\}$
17	$(error, closed)$	cl	$\{fob, fos\}$
18	$(error, closed)$	ε	$\{fcb, fos\}$
19	$(error, open)$	ε	$\{fcb, fos\}$
20	$(awake, open)$	ε	$\{fob\}$
21	$(error, closed)$	ε	$\{fob, fos\}$
22	$(idle, open)$	cl	$\{fob\}$
23	$(error, open)$	op	$\{fob, fcs\}$
24	$(idle, open)$	ε	$\{fob, fcb\}$
25	$(error, open)$	cl	$\{fob, fcb, fos\}$
26	$(error, open)$	ε	$\{fob, fcb, fos\}$
27	$(awake, open)$	op	$\{fcb\}$
28	$(awake, open)$	ε	$\{fcb\}$
29	$(idle, open)$	cl	$\{fcb\}$
30	$(error, open)$	op	$\{fcb, fcs\}$
31	$(idle, closed)$	ε	$\{fcb\}$
32	$(error, open)$	ε	$\{fcb, fcs\}$
33	$(awake, closed)$	op	$\{fcb\}$
34	$(error, closed)$	cl	$\{fcb, fos\}$
35	$(awake, closed)$	ε	$\{fob, fcb\}$
36	$(idle, closed)$	cl	$\{fob, fcb\}$
37	$(error, closed)$	op	$\{fob, fcb, fcs\}$
38	$(error, open)$	ε	$\{fob, fcb, fcs\}$
39	$(error, closed)$	ε	$\{fob, fcb, fcs\}$
40	$(error, closed)$	ε	$\{fob, fcb, fos\}$
41	$(error, closed)$	cl	$\{fob, fcb, fos\}$
42	$(idle, closed)$	ε	$\{fob, fcb\}$
43	$(awake, closed)$	op	$\{fob, fcb\}$
44	$(awake, open)$	ε	$\{fob, fcb\}$
45	$(idle, open)$	cl	$\{fob, fcb\}$
46	$(error, open)$	op	$\{fob, fcb, fcs\}$
47	$(awake, open)$	op	$\{fob, fcb\}$

Fig. 2. Diagnosis space $\mathcal{P}^*$ (left) and relevant state details (right).

Example 3. With reference to the DES $\mathcal{P}$ introduced in Example 1 (cf. Figure 1 and Table 1), outlined on the left side of Figure 2 is the diagnosis space $\mathcal{P}^*$, where states are identified by numbers $0\,..\,47$; state details are listed in the table on the right side. Specifically, each state $p^* \in \mathcal{P}^*$ is represented by a triple (C, L, δ), where C is the pair of states of the sensor s and the breaker b, L is the (possibly empty) event within the link, and δ is the diagnosis associated with p^*. Owing to cycles, the set of possible trajectories of $\mathcal{P}$ is infinite. One of these trajectories is $[s_1, b_1, s_2, b_4]$, ending in the state $10 = ((idle, open), \varepsilon, \{fcb\})$, which corresponds to the following events: s detects a threatening event and commands b to open; b opens; s detects a liberating event and commands b to close; still, b remains open. In fact, the diagnosis $\{fcb\}$ accounts for the failing of the breaker to close.

When diagnosis is performed *online*, while the DES is being monitored, some sort of diagnosis information is expected at the occurrence of each observation. This is captured by the notion of a *temporal explanation*.

Definition 1. *Let $\mathcal{O} = [o_1, \ldots, o_n]$ be a symptom of $\mathcal{X}$ and T a trajectory of $\mathcal{X}$ implying $\mathcal{O}$. Let $\mathcal{O}_{[i]}$, $i \in [0\,..\,n]$, denote the prefix of $\mathcal{O}$ up to o_i. Let $T_{[i]}$, $i \in [0\,..\,n]$, denote either T, if $i = n$, or the prefix of T up to the transition preceding the $(i+1)$-th observable transition in T, if $0 \leq i < n$. The* temporal explanation *of $\mathcal{O}$ is the sequence of sets of* candidate diagnoses*, $\boldsymbol{\Delta}(\mathcal{O}) = [\Delta_0, \Delta_1, \ldots, \Delta_n]$, where each Δ_i, $i \in [0\,..\,n]$, is the minimal set of diagnoses defined as follows:*

$$T \in \mathcal{X}^*, \forall i \in [0\,..\,n] \left(\Delta_i \supseteq \left\{\delta(T_{[i]}) \mid T_{[i]} \Rightarrow \mathcal{O}_{[i]}\right\}\right). \tag{4}$$

Example 4. Let $\mathcal{O} = [act, opn, sby, act, cls]$ be a symptom of the DES $\mathcal{P}$. As such, $\mathcal{O}$ is implied by just one trajectory, namely $T = [s_1, b_1, s_2, b_4, s_3, b_2]$. Thus, $\boldsymbol{\Delta}(\mathcal{O}) = [\Delta_0, \Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5]$, where $\Delta_0 = \Delta_1 = \Delta_2 = \{\emptyset\}$, $\Delta_3 = \{\{fcb\}\}$, and $\Delta_4 = \Delta_5 = \{\{fcb, fos\}\}$.

When a DES is being monitored, the temporal explanation needs to be updated at the occurrence of each newly generated observation, as the symptom of the DES is not output in one shot, but one observation at a time. Still, updating the temporal explanation at the occurrence of the observation o_{i+1} does not boil down to simply extending $\boldsymbol{\Delta}(\mathcal{O}_{[i]})$ by Δ_{i+1}. Instead, generally speaking, each Δ_j, $j \leq i$, needs to be updated (specifically, pruned).

Example 5. With reference to Example 4, the temporal explanation of $\mathcal{O}$ after the third observation, is $\boldsymbol{\Delta}([act, opn, sby]) = [\emptyset, \emptyset, \emptyset, \{\emptyset, \{fcb\}, \{fcs\}\}]$. The set $\Delta_3 = \{\emptyset, \{fcb\}, \{fcs\}\}$ includes the diagnoses relevant to the trajectories ending in the states 6, 7, 10, or 11 of $\mathcal{P}^*$ (cf. Figure 2). However, after the reception of the fourth observation, namely act, only the transition up to the state 10 is consistent (being exited by s_1 and s_3), which implies the diagnosis $\{fcb\}$. Hence, the other two candidates in Δ_3, namely $\emptyset$ and $\{fcs\}$, are removed, so that the extended temporal explanation becomes $\boldsymbol{\Delta}([act, opn, sby, act]) = [\emptyset, \emptyset, \emptyset, \{\{fcb\}\}, \{\{fcb\}, \{fcb, fos\}\}]$, where $\Delta_4 = \{\{fcb\}, \{fcb, fos\}\}$ is the set of the diagnoses implied by the trajectories ending in states 15, 19, 27, or 28.

4 Temporal Dictionary

A technique for preprocessing a DES $\mathcal{X}$ in order to generate a DFA, the *temporal dictionary*, for supporting the online diagnosis of $\mathcal{X}$ efficiently is presented.

Definition 2. *Let $\mathcal{X}^*$ be a diagnosis space. Let $\mathcal{X}^*_{\mathrm{n}}$ be the nondeterministic finite automaton (NFA) obtained from $\mathcal{X}^*$ by substituting the observation o for the component transition t marking each transition in $\mathcal{X}^*$, where $(t, o, f) \in \mu(\mathcal{X})$. The* temporal dictionary *of $\mathcal{X}$ is the DFA $\mathcal{X}^{\circledast}$ obtained by the determinization of $\mathcal{X}^*_{\mathrm{n}}$, which is decorated by the following additional information:*

1. *Each state $x^{\circledast}$ in $\mathcal{X}^{\circledast}$ is marked with the sets $\lfloor x^{\circledast} \rfloor$, $\|x^{\circledast}\|$, and $\Delta(x^{\circledast})$, where:*
   (a) *$\lfloor x^{\circledast} \rfloor$ is the set of states of $\mathcal{X}^*_{\mathrm{n}}$ included in $x^{\circledast}$,*[2]
   (b) *$\|x^{\circledast}\|$ is the set of pairs (x^*_1, x^*_2) where $x^*_1 \in \lfloor x^{\circledast} \rfloor$, $x^*_2 \in \lfloor x^{\circledast} \rfloor$, x^*_1 is entered by an observable transition in $\mathcal{X}^*_{\mathrm{n}}$, x^*_2 is exited by an observable transition in $\mathcal{X}^*_{\mathrm{n}}$, and there is a (possibly empty) sequence of ε-transitions in $\mathcal{X}^*_{\mathrm{n}}$ connecting x^*_1 with x^*_2,*
   (c) *$\Delta(x^{\circledast})$ is the set of diagnoses associated with the $\mathcal{X}^*_{\mathrm{n}}$ states in $\lfloor x^{\circledast} \rfloor$;*
2. *Each transition $\langle x^{\circledast}_1, o, x^{\circledast}_2 \rangle$ in $\mathcal{X}^{\circledast}$ is marked with $\lfloor \langle x^{\circledast}_1, o, x^{\circledast}_2 \rangle \rfloor$, denoting the set of transitions $\langle x^*_1, o, x^*_2 \rangle$ in $\mathcal{X}^*_{\mathrm{n}}$ where $x^*_1 \in \lfloor x^{\circledast}_1 \rfloor$ and $x^*_2 \in \lfloor x^{\circledast}_2 \rfloor$.*

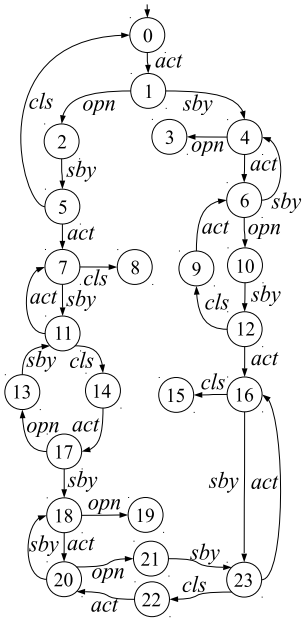
Proposition 1. *The language of $\mathcal{X}^{\circledast}$ equals the set of possible symptoms of $\mathcal{X}$. Besides, if $\mathcal{O}$ is a symptom in $\mathcal{X}^{\circledast}$ with accepting state $x^{\circledast}$, then $\Delta(x^{\circledast}) = \Delta(\mathcal{O})$.*

Remarkably, according to Proposition 1, the explanation of a symptom $\mathcal{O}$ is materialized in the state of the temporal dictionary accepting the string $\mathcal{O}$, hence making the generation of the explanation of $\mathcal{O}$ very efficient.

Example 6. With reference to the diagnosis space $\mathcal{P}^*$ in Figure 2, the temporal dictionary $\mathcal{P}^{\circledast}$ is outlined on the left side of Figure 3, with explanations being listed in the table shown on the right side. Details of states and transitions are displayed in Figure 4. Specifically, each (shadowed) state $p^{\circledast}$ incorporates the relevant set $\lfloor p^{\circledast} \rfloor$ of $\mathcal{P}^*$ states, along with the set of connections $\|p^{\circledast}\|$, indicated by internal arcs. Each transition $\langle p^{\circledast}, o, {p'}^{\circledast} \rangle$ is unfolded into the set of transitions $\lfloor \langle p^{\circledast}, o, {p'}^{\circledast} \rangle \rfloor$ in $\mathcal{P}^*$. In accordance with Proposition 1, given $\mathcal{O} = [act, opn, sby, act, cls]$, which has accepting state 8 in $\mathcal{P}^{\circledast}$, we have $\Delta(8) = \{\{fcb, fos\}\} = \Delta(\mathcal{O})$ (cf. Example 4).

Based on Definition 1, the temporal explanation of $\mathcal{O} = [o_1, \ldots, o_n]$ is a sequence $\boldsymbol{\Delta}(\mathcal{O}) = [\Delta_0, \Delta_1, \ldots, \Delta_n]$ where each Δ_i, $i \in [0\,..\,n]$, is the set of diagnoses implied by $T_{[i]}$, where $T_{[i]}$ also implies $\mathcal{O}_{[i]}$. Here, the key point is that the same trajectory T must fulfill these conditions for *all* prefixes $\mathcal{O}_{[i]}$. This property makes a temporal explanation consistent: for each diagnosis $\delta_i \in \Delta_i$, $i \in [0\,..\,(n-1)]$, there is a diagnosis $\delta_{i+1} \in \Delta_{i+1}$ such that $\delta_i \subseteq \delta_{i+1}$, and vice versa.

[2] According to the *Subset Construction* determinization algorithm [7], each state of the DFA is identified by a subset of the states of the NFA.



State $p^{\circledast}$	Explanation $\Delta(p^{\circledast})$
0	$\{\emptyset\}$
1	$\{\emptyset, \{fob\}, \{fos\}\}$
2	$\{\emptyset\}$
3	$\{\{fob, fcs\}\}$
4	$\{\{fob\}, \{fob, fcs\}\}$
5	$\{\emptyset, \{fcb\}, \{fcs\}\}$
6	$\{\{fob\}, \{fob, fos\}\}$
7	$\{\{fcb\}, \{fcb, fos\}\}$
8	$\{\{fcb, fos\}\}$
9	$\{\{fob\}\}$
10	$\{\{fob\}\}$
11	$\{\{fcb\}, \{fcb, fcs\}\}$
12	$\{\{fob\}, \{fob, fcb\}, \{fob, fcs\}\}$
13	$\{\{fcb\}\}$
14	$\{\{fcb\}\}$
15	$\{\{fob, fcb, fos\}\}$
16	$\{\{fob, fcb\}, \{fob, fcb, fos\}\}$
17	$\{\{fcb\}, \{fob, fcb\}, \{fcb, fos\}\}$
18	$\{\{fob, fcb\}, \{fob, fcb, fcs\}\}$
19	$\{\{fob, fcb, fcs\}\}$
20	$\{\{fob, fcb\}, \{fob, fcb, fos\}\}$
21	$\{\{fob, fcb\}\}$
22	$\{\{fob, fcb\}\}$
23	$\{\{fob, fcb\}, \{fob, fcb, fcs\}\}$

Fig. 3. Temporal dictionary $\mathcal{P}^{\circledast}$ (left) and relevant explanations (right).

Example 7. Consider the symptom $\mathcal{O} = [act, opn, sby, act, cls]$ in regard to the temporal dictionary $\mathcal{P}^{\circledast}$ outlined in Figure 3. The accepting states of $\mathcal{O}_{[i]}$, $i \in [0\,..\,5]$, are $0, 1, 2, 5, 7$, and 8, respectively, which are marked with the sets of diagnoses $\Delta(0) = \{\emptyset\}$, $\Delta(1) = \{\emptyset, \{fob\}, \{fos\}\}$, $\Delta(2) = \{\emptyset\}$, $\Delta(5) = \{\emptyset, \{fcb\}, \{fcs\}\}$, $\Delta(7) = \{\{fcb\}, \{fcb, fos\}\}$, and $\Delta(8) = \{\{fcb, fos\}\}$. However, the sequence $[\Delta(0),\ \Delta(1), \Delta(2), \Delta(5), \Delta(7), \Delta(8)]$ does *not* expand consistently and, hence, can *not* be the temporal explanation of $\mathcal{O}$.

Example 7 clearly shows that the temporal explanation of $\mathcal{O}$ cannot simply be the sequence of the explanations of each $\mathcal{O}_{[i]}$, because only a subset of the trajectories implying $\mathcal{O}_{[i]}$ are in general prefixes of the trajectories implying $\mathcal{O}_{[i+1]}$, as a consequence of the constraints imposed by the new observation o_{i+1}.

Example 8. With reference to Example 7, we have $[\Delta(0), \Delta(1), \Delta(2)] = [\{\emptyset\}, \{\emptyset, \{fob\}, \{fos\}\}, \{\emptyset\}]$. Clearly, $\Delta(2) = \{\emptyset\}$ is *not* a consistent expansion of $\Delta(1) = \{\emptyset, \{fob\}, \{fos\}\}$. In fact, considering $\mathcal{P}^{\circledast}$ outlined in Figure 4, initially we have $\Delta_0 = \Delta(0) = \{\emptyset\}$. At the reception of the first observation act, the accepting state in $\mathcal{P}^{\circledast}$ is 1, thereby $\Delta(1)$ is the set of diagnoses associated with the states within 1, namely $\{\emptyset, \{fob\}, \{fos\}\}$. At the reception of the second observation opn, the accepting state becomes 2, including just the $\mathcal{X}^*$ state $3 = ((awake, open), \varepsilon, \emptyset)$. Hence, we have $\Delta(2) = \{\emptyset\}$. The point is, after the occurrence of the observation opn, as clearly indicated in Figure 4, the only trajectory in $\mathcal{P}^*$ that is consistent with $[act, opn]$ is $0 \to 1 \to 3$. Consequently, the

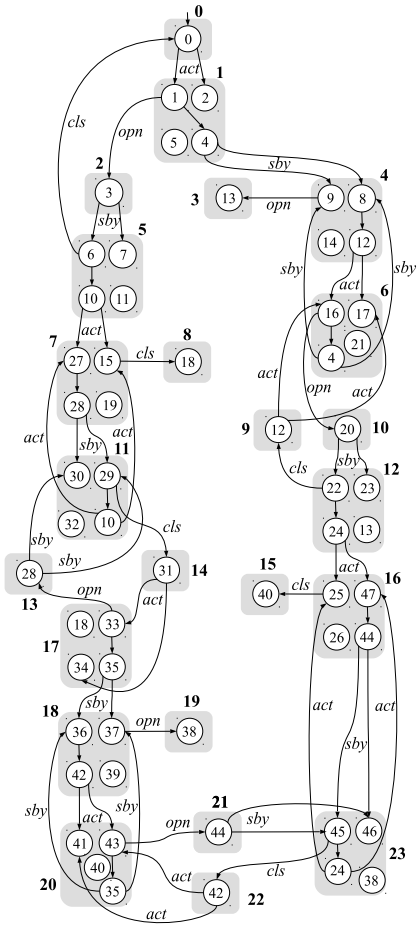


Fig. 4. Details of the temporal dictionary $\mathcal{P}^{\circledast}$ outlined in Figure 3.

candidate diagnoses associated with the $\mathcal{P}^*$ states 2, 4, and 5 in the state 1 need to be removed from Δ_1, thereby obtaining $[\Delta_0, \Delta_1, \Delta_2] = [\emptyset, \emptyset, \emptyset]$, which, based on Definition 1, is in fact the temporal explanation of $[act, opn]$. In the worst case, this pruning needs to be propagated backward in $\boldsymbol{\Delta}(\mathcal{O})$ up to Δ_0.

When a DES $\mathcal{X}$ is being operated, a symptom of $\mathcal{X}$ is generated *one observation at a time*. Assuming that the current symptom is $\mathcal{O}_{[i]} = [o_1, \ldots, o_i]$ and the corresponding temporal explanation $\boldsymbol{\Delta}(\mathcal{O}_{[i]})$ has been generated already, the occurrence of a new observation o_{i+1} requires the diagnosis engine to expand the temporal explanation by the insertion of Δ_{i+1} and, in the worst case, by the backward pruning of $\Delta_i, \Delta_{i-1}, \ldots, \Delta_0$, thereby generating the temporal explanation $\boldsymbol{\Delta}(\mathcal{O}_{[i+1]})$. In order to perform this task efficiently, each diagnosis set Δ_i

in the temporal explanation is associated with additional information, leading to the notion of a *temporal abduction* (Definition 3).

Definition 3. *Let* $\mathcal{O} = [o_1, \ldots, o_n]$ *be a symptom of* $\mathcal{X}$*. The* temporal abduction $\mathcal{A}$ *of* $\mathcal{O}$ *is a sequence* $\mathcal{A}(\mathcal{O}) = [\alpha_0, \alpha_1, \ldots, \alpha_n]$*, where* $\forall i \in [0\,..\,n]$*,* $\alpha_i = \left(X_i^*, x_i^{\circledast}, \Delta_i\right)$*, where* $x_i^{\circledast}$ *is the accepting state of* $\mathcal{O}_{[i]}$ *in* $\mathcal{X}^{\circledast}$*, while* $X_i^* \subseteq \lfloor x_i^{\circledast} \rfloor$ *and* $\Delta_i \subseteq \Delta\left(x_i^{\circledast}\right)$ *are defined by the following rules:*

1. $X_0^* = \emptyset$*;*
2. $\Delta_n = \Delta\left(x_n^{\circledast}\right)$*;*
3. *If* $n \neq 0$*, then* $X_n^* = \left\{x_n^* \mid (x_{n-1}^*, o_n, x_n^*) \in \lfloor (x_{n-1}^{\circledast}, o_n, x_n^{\circledast}) \rfloor\right\}$*;*
4. *If* $i < n$*, then* $\Delta_i = \{\delta_i \mid x_i^* = (C_i, L_i, \delta_i), \langle x_i^*, o_i, x_{i+1}^* \rangle \in \lfloor \langle x_i^{\circledast}, o_i, x_{i+1}^{\circledast} \rangle \rfloor,$ $x_{i+1}^* \in X_{i+1}^*\}$*;*
5. *If* $i \neq 0$ *and* $i \neq n$*, then* $X_i^* = \{x_i^* \mid (x_i^*, x_i'^*) \in \left\|x_i^{\circledast}\right\|, \langle x_i'^*, o_i, x_{i+1}^* \rangle \in$ $\lfloor \langle x_i^{\circledast}, o_i, x_{i+1}^{\circledast} \rangle \rfloor,\ x_{i+1}^* \in X_{i+1}^*\}$*.*

That is, the temporal abduction of $\mathcal{O}$ is a sequence of triples $\left(X_i^*, x_i^{\circledast}, \Delta_i\right)$, where $x_i^{\circledast}$ is the accepting state of $\mathcal{O}_{[i]}$ in the temporal dictionary $\mathcal{X}^{\circledast}$, Δ_i happens to equal the homonymous element in $\boldsymbol{\Delta}(\mathcal{O})$ (cf. Proposition 2 below), and X_i^* is the set of $\mathcal{X}^*$ states that are entered by the trajectories fulfilling Definition 1. When a new observation occurs while monitoring $\mathcal{X}$, the set X_i^* is key to backward pruning the temporal abduction, as illustrated in the next example.

Example 9. With reference to the temporal dictionary $\mathcal{P}^{\circledast}$ (Figures 3 and 4), consider the symptom $\mathcal{O}$ defined in Example 7, where $\mathcal{O}_{[4]} = [act, opn, sby, act]$. Based on Definition 3, we have $\mathcal{A}(\mathcal{O}_{[4]}) = [(\emptyset, 0, \{\emptyset\}), (\{1\}, 1, \{\emptyset\}), (\{3\}, 2, \{\emptyset\}),$ $(\{6\}, 5, \{\{fcb\}\}), (\{15, 27\}, 7, \{\{fcb\}, \{fcb, fos\}\})]$, where the diagnosis set in the last triple incorporates the diagnoses marking the $\mathcal{P}^*$ states 15, 19, 27, and 28 (those included in the state 7 of $\mathcal{P}^{\circledast}$). Then, assume the occurrence of the new observation *cls*, leading to the accepting state 8 of $\mathcal{P}^{\circledast}$. Based on Definition 3, we have (rule 2) $\Delta_5 = \Delta(8) = \{\{fcb, fos\}\}$ and (rule 3) $X_5^* = \{18\}$. In other words, $\alpha_5 = (\{18\}, 8, \{\{fcb, fos\}\})$. Now, backward pruning starts. First, based on X_5^*, we get (rules 4) $\Delta_4 = \{\{fcb, fos\}\}$, where $\{fcb, fos\}$ is associated with the state 15 and the diagnosis $\{fcb\}$ has been removed, and (rule 5) $X_4^* = \{15\}$, where the state 27 has been removed. At this point, the application of the same rules based on X_4^* has no effect on α_3: since $X_3^* = \{6\}$ is unchanged, the backward pruning stops and eventually $\mathcal{A}(\mathcal{O}_{[5]})$ fulfills Definition 3.

Proposition 2. *If* $\mathcal{O}$ *is a symptom of* $\mathcal{X}$*, then*

$$\boldsymbol{\Delta}(\mathcal{O}) = \left[\, \Delta_i \mid (X_i^*, x_i^{\circledast}, \Delta_i) \in \mathcal{A}(\mathcal{O}) \,\right]. \tag{5}$$

Based on Proposition 2, the temporal explanation $\boldsymbol{\Delta}(\mathcal{O})$ can be generated as a projection of the abduction $\mathcal{A}(\mathcal{O})$. The temporal explanation is required to be generated while the DES is being monitored: starting from the initial diagnosis set Δ_0, which corresponds to the empty symptom, the temporal explanation is constructed one observation at a time. Assuming that the temporal explanation $\boldsymbol{\Delta}(\mathcal{O}_{[i]})$ of the current symptom up to the i-th observation is available, the occurrence of the observation o_{i+1} requires the generation of $\boldsymbol{\Delta}(\mathcal{O}_{[i+1]})$.

Algorithm 1 *Abduce*

1: **procedure** ABDUCE($\mathcal{X}^{\circledast}$, $\mathcal{O}_{[i]}$, $\mathcal{A}$, o_{i+1})
2: $\mathcal{X}^{\circledast} = \left(\Sigma, X^{\circledast}, \tau^{\circledast}, x_0^{\circledast}\right)$: the temporal dictionary of $\mathcal{X}$
3: $\mathcal{O}_{[i]} = [o_1, \ldots, o_i]$: the prefix of a symptom of $\mathcal{X}$ up to the i-th observation
4: $\mathcal{A} = [\alpha_0, \alpha_1, \ldots, \alpha_i]$: the abduction of $\mathcal{O}_{[i]}$
5: o_{i+1}: the next observation of $\mathcal{X}$
6: **begin**
7: Let $x_i^{\circledast}$ be the state of $\mathcal{X}^{\circledast}$ in the triple α_i
8: $x_{i+1}^{\circledast} \leftarrow \tau^{\circledast}(x_i^{\circledast}, o_{i+1})$
9: $\Delta_{i+1} \leftarrow \Delta(x_{i+1}^{\circledast})$
10: $X_{i+1}^{*} \leftarrow \left\{x_{i+1}^{*} \mid (x_i^{*}, o_{i+1}, x_{i+1}^{*}) \in \lfloor (x_i^{\circledast}, o_{i+1}, x_{i+1}^{\circledast}) \rfloor\right\}$
11: Extend $\mathcal{A}$ by the new triple $\alpha_{i+1} = \left(X_{i+1}^{*}, x_{i+1}^{\circledast}, \Delta_{i+1}\right)$
12: **for all** j **from** i **downto** 0 **do** # *Backward pruning of the temporal abduction*
13: Let $\alpha_j = (X_j^{*}, o_j, \Delta_j)$ be the j-th triple in $\mathcal{A}$
14: $\Delta_{\text{new}} \leftarrow \left\{\delta_j \mid x_j^{*} = (x_j, \delta_j), (x_j^{*}, o_j, x_{j+1}^{*}) \in \lfloor (x_j^{\circledast}, o_j, x_{j+1}^{\circledast}) \rfloor, x_{j+1}^{\circledast} \in X_{j+1}^{*}\right\}$
15: **if** $\Delta_{\text{new}} \neq \Delta_j$ **then**
16: Substitute Δ_{new} for Δ_j in α_j
17: **end if**
18: **if** $i \neq 0$ **then** # *Based on rule 1 of Definition 3, the value of* X_0^{*} *is fixed to* $\emptyset$
19: $X_{\text{new}}^{*} \leftarrow \left\{x_j^{*} \mid (x_j^{*}, x_j'^{*}) \in \left\|x_j^{\circledast}\right\|, (x_j'^{*}, o_j, x_{j+1}^{*}) \in \lfloor (x_j^{\circledast}, o_j, x_{j+1}^{\circledast}) \rfloor, x_{j+1}^{*} \in X_{j+1}^{*}\right\}$
20: **if** $X_{\text{new}}^{*} \neq X_j^{*}$ **then**
21: Substitute X_{new}^{*} for X_j^{*} in α_j
22: **else**
23: **break** # *If* X_{new}^{*} *equals* X_j^{*}*, then backward pruning has no effect*
24: **end if**
25: **end if**
26: **end for**
27: **end procedure**

5 The *Abduce* Algorithm

In operational terms, the generation of the temporal explanation is specified by the *Abduce* algorithm (Algorithm 1, lines 1–27). Given the temporal dictionary $\mathcal{X}^{\circledast}$, the current symptom $\mathcal{O}_{[i]}$, the corresponding temporal abduction $\mathcal{A}$, and the new observation o_{i+1}, the algorithm updates $\mathcal{A}$ to obtain the temporal abduction of $\mathcal{O}_{[i+1]}$. To this end, the accepting state $x_{i+1}^{\circledast}$ of $\mathcal{O}_{[i+1]}$ is determined (lines 7 and 8). Then, based on the rules 3 and 4 of Definition 3, Δ_{i+1} and X_{i+1}^{*} are generated (lines 9 and 10), thus allowing for the construction of the new triple α_{i+1} (line 11). Backward pruning is performed in lines 12–26. Each triple α_j, with j ranging from i down to 0, is updated based on the rules 4 and 5 of Definition 3. However, this pruning may stop before the natural end of the loop, namely when X_{new}^{*} equals X_j^{*} (line 23). If this condition holds, then, based on Definition 3, all the triples $\alpha_0, \ldots, \alpha_{j-1}^{*}$ keep the same value.

Example 10. With reference to Figure 3 and Figure 4, consider the temporal dictionary $\mathcal{P}^{\circledast}$. Let $\mathcal{O} = [act, opn, sby, act, cls]$ be a symptom of $\mathcal{P}$. Traced in Table 2 is the generation of the temporal abduction $\mathcal{A}(\mathcal{O})$, one observation at

Table 2. Incremental generation of $\mathcal{A}([act, opn, sby, act, cls])$ by the *Abduce* algorithm.

i	α_0	α_1	α_2	α_3	α_4	α_5
0	$(\emptyset, 0, \{\emptyset\})$					
1	$(\emptyset, 0, \{\emptyset\})$	$(\{1,2\}, 1, \{\emptyset, \{fob\}, \{fos\}\})$				
2	$(\emptyset, 0, \{\emptyset\})$	$(\{1,2\}, 1, \{\emptyset, \cancel{\{fob\}}, \cancel{\{fos\}}\})$	$(\{3\}, 2, \{\emptyset\})$			
3	$(\emptyset, 0, \{\emptyset\})$	$(\{1\}, 1, \{\emptyset\})$	$(\{3\}, 2, \{\emptyset\})$	$(\{6,7\}, 5, \{\emptyset, \{fcb\}, \{fcs\}\})$		
4	$(\emptyset, 0, \{\emptyset\})$	$(\{1\}, 1, \{\emptyset\})$	$(\{3\}, 2, \{\emptyset\})$	$(\{6,\cancel{7}\}, 5, \{\emptyset, \{fcb\}, \cancel{\{fcs\}}\})$	$(\{15, 27\}, 7, \{\{fcb\}, \{fcb, fos\}\})$	
5	$(\emptyset, 0, \{\emptyset\})$	$(\{1\}, 1, \{\emptyset\})$	$(\{3\}, 2, \{\emptyset\})$	$(\{6\}, 5, \{\{fcb\}\})$	$(\{15, \cancel{27}\}, 7, \{\cancel{\{fcb\}}, \{fcb, fos\}\})$	$(\{18\}, 8, \{\{fcb, fos\}\})$

a time, where pruning is denoted by strike-through. Each row i of the table represents the configuration of the temporal abduction after the reception of the i-th observation. Initially ($i = 0$), based on rules 1 and 2 of Definition 3, we have $\alpha_0 = (\emptyset, 0, \{\emptyset\})$. Upon the reception of $o_1 = act$, according to Algorithm 1, the accepting state is $x_1^{\circledast} = 1$, thereby $\Delta_1 = \Delta(1) = \{\emptyset, \{fob\}, \{fos\}\}$ and $\mathcal{X}_1^* = \{1, 2\}$. Backward pruning has no effect. Upon the reception of $o_2 = opn$, the new triple is $\alpha_2 = (\{3\}, 2, \{\emptyset\})$. In this case, backward pruning removes the candidates $\{fob\}$ and $\{fos\}$ from Δ_1, as only the transition $\langle 1, opn, 3\rangle$ in X_1^* is involved (cf. Figure 4). However, since X_1^* is unchanged, no further pruning is applied. The reception of $o_3 = sby$ moves to the accepting state $x_3^{\circledast} = 5$, thereby creating the new triple $\alpha_3 = (\{6, 7\}, 5, \{\emptyset, \{fcb\}, \{fcs\}\})$, without backward pruning. The reception of $o_4 = act$ moves to the accepting state $x_4^{\circledast} = 7$, thereby creating the new triple $\alpha_4 = (\{15, 27\}, 7, \{\{fcb\}, \{fcb, fos\}\})$. Since the involved transition exits state 10 in $x_3^{\circledast} = 5$, Δ_3 is reduced to $\{\{fcb\}\}$, where $\{fcb\}$ is the diagnosis marking the state 10. Furthermore, the state 7 is removed from X_3^*, because it is not connected with 10. No further pruning is applicable. Finally, after the reception of the last observation $o_5 = cls$, the accepting state is $x_5^{\circledast} = 8$, thereby generating the triple $\alpha_5 = (\{18\}, 8, \{\{fcb, fos\}\})$. Since the involved transition exits the state 15 in $x_4^{\circledast} = 7$, Δ_4 is reduced to $\{\{fcb, fos\}\}$. Also, the state 27 is removed from X_4^*. No other pruning is applicable. Eventually, according to Proposition 2, we have $\boldsymbol{\Delta}(\mathcal{O}) = [\{\emptyset\}, \{\emptyset\}, \{\emptyset\}, \{\{fcb\}\}, \{\{fcb, fos\}\}, \{\{fcb, fos\}\}]$, which in fact equals the temporal explanation determined in Example 4 based on Definition 1. The sequence clearly shows to an operator in charge of monitoring the system (and possibly of performing recovery actions) that no fault occurred up to the second observation; then, two faults occurred in cascade, namely *fcb* and *fos*. Without backward pruning, the list of sets of candidate diagnoses is $[\{\emptyset\}, \{\emptyset, \{fob\}, \{fos\}\}, \{\emptyset\}, \{\emptyset, \{fcb\}, \{fcs\}\}, \{\{fcb\}, \{fcb, fos\}\}, \{\{fcb, fos\}\}]$, the interpretation of which may be misleading to the operator. After all, the latter is *not* the temporal explanation of $\mathcal{O}$.

6 Dual Knowledge Compilation

A temporal dictionary $\mathcal{X}^{\circledast}$ is an extremely efficient tool for supporting the diagnosis of DESs. In theory, the temporal dictionary allows the DE to operate always in quick mode by *Engine 1*, with *Engine 2* never coming into play. However, the

temporal dictionary requires total knowledge compilation, which is out of dispute for practical reasons. So, in order to escape from total knowledge compilation and somewhat retaining the advantage of *Engine 1*, we propose a restricted dictionary that expands over time either by experience or by the injection of specific knowledge. In other words, we propose *dual knowledge compilation* based on an *open dictionary*. Intuitively, an open dictionary is a subgraph of the temporal dictionary whose language is a subset of the language of the temporal dictionary (the set of symptoms of the DES). As such, each symptom $\mathcal{O}$ in the open dictionary is associated with $\Delta(\mathcal{O})$, the *sound and complete* set of candidate diagnoses that explain $\mathcal{O}$. Hence, despite being sound (but not complete) in the set of symptoms, the open dictionary is sound and complete in the explanation of the symptom, provided that the symptom is included in the language of the open dictionary. What is the initial configuration of the open dictionary? We suggest to initialize the open dictionary with a *prefix* of the temporal dictionary, as specified in Definition 4.

Definition 4. *Let $\mathcal{X}^{\circledast}$ be a temporal dictionary. The* distance *of a state $x^{\circledast}$ in $\mathcal{X}^{\circledast}$ is the minimum number of transitions connecting the initial state of $\mathcal{X}^{\circledast}$ with $x^{\circledast}$. The* prefix *of $\mathcal{X}^{\circledast}$ up to a distance $d \geq 0$, denoted $\mathcal{X}^{\circledast}_{[d]}$, is the subgraph of $\mathcal{X}^{\circledast}$ comprehending all the states at distance $\leq d$ and all the transitions exiting the states at distance $< d$.*

Example 11. With reference to Figure 3, the prefix of $\mathcal{P}^{\circledast}$ up to distance 2, namely $\mathcal{P}^{\circledast}_{[2]}$, is displayed on the left side of Figure 5.

A prefix $\mathcal{X}^{\circledast}_{[d]}$ provides the explanation of every symptom that is not longer than d. If $\mathcal{X}^{\circledast}_{[d]}$ embodies a cycle (which is not the case in $\mathcal{P}^{\circledast}_{[2]}$), it also provides the explanation of the infinite set of symptoms encompassing this cycle. However, any symptom longer than d may not belong to the language of $\mathcal{X}^{\circledast}_{[d]}$, such as $\mathcal{O} = [act, sby, opn]$ in $\mathcal{P}^{\circledast}_{[2]}$. In this case, comes into play *Engine 2*, which generates the temporal explanation $\Delta(\mathcal{O})$ based on the *abduction* of $\mathcal{O}$, namely a DFA whose language is the subset of the trajectories of $\mathcal{X}$ implying $\mathcal{O}$. To this end, *Engine 2* performs model-based reasoning to reconstruct the subspace of $\mathcal{X}^{\circledast}$ required. Once provided the temporal explanation $\Delta(\mathcal{O})$, the experience acquired by the DE can be integrated into the open dictionary based on the *symptom*

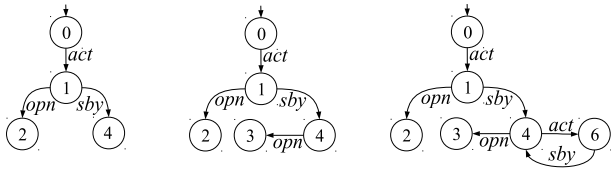


Fig. 5. From left to right, expansion of the open dictionary: $\mathcal{P}^{\circledast}_{[2]}$, $\mathcal{P}^{\circledast}_{[2,\mathcal{O}]}$, and $\mathcal{P}^{\circledast}_{[2,\mathcal{O},\mathcal{S}]}$.

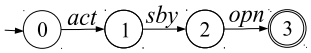


Fig. 6. (Simple) symptom pattern $\mathcal{O}^*$, where $\mathcal{O} = [act, sby, opn]$.

pattern of $\mathcal{O}$. However, the notion of a symptom pattern goes beyond a (plain) symptom, as specified below.

Definition 5. *A* symptom pattern *of a DES $\mathcal{X}$ is a DFA whose language is a subset of the symptoms of $\mathcal{X}$.*

A special (and very simple) case of symptom pattern is associated with each symptom $\mathcal{O}$, denoted $\mathcal{O}^*$, which is the DFA recognizing $\mathcal{O}$ (the language of $\mathcal{O}^*$ is the singleton $\{\mathcal{O}\}$).

Example 12. Displayed in Figure 6 is the symptom pattern of $\mathcal{O} = [act, sby, opn]$. Another (circular) symptom pattern is displayed on the bottom-right side of Figure 7 (cf. Example 16).

Given a symptom pattern $\mathcal{O}^*$, the language of an open dictionary $\mathcal{X}^{\circledast}$ can be extended by the language of $\mathcal{O}^*$ by means of the *Dictionary Extension* algorithm listed below (cf. Algorithm 2, lines 1–31). We assume that each state $x^{\circledast}$ in $\mathcal{X}^{\circledast}$ is equipped with a *labeling set*, denoted $\Omega(x^{\circledast})$ (initially empty), which is instantiated with states in $\mathcal{O}^*$. The algorithm aims to match $\mathcal{O}^*$ with the language of $\mathcal{X}^{\circledast}$. When the matching of an observation o succeeds, the labeling set of the state reached in $\mathcal{X}^{\circledast}$ is extended by the state reached in $\mathcal{O}^*$(provided it is not included). If the matching fails, then $\mathcal{X}^{\circledast}$ is extended by a new transition and, possibly, by a new state. Let $\langle x^{\circledast}, o, x'^{\circledast}\rangle$ be the missing transition in $\mathcal{X}^{\circledast}$. Based on lines 13–15, the new state $x'^{\circledast}$ is generated first by determining the set of $\mathcal{X}^*$ states exited by a transition marked by o and, then, by extending this set with all the transitions exiting these states that are marked by an unobservable component transition. It should be clear that $\mathcal{X}^*$ is not materialized: only the states required are actually generated starting from the $\mathcal{X}^*$ states within $x^{\circledast}$ and stored in $\mathcal{X}^{\circledast}$. Once all the transitions exiting the state $\omega \in \mathcal{O}^*$ considered have been processed, $\omega \in \Omega(x^{\circledast})$ is marked (line 27). Given $\omega \in \Omega(x^{\circledast})$, two cases are possible. If ω is not marked, then the transition function of $x^{\circledast}$ in $\mathcal{X}^{\circledast}$ needs to be checked against $\mathcal{O}^*$. If, instead, ω is marked, then the update of the transition function of $x^{\circledast}$ is completed. Hence, since it is impossible to insert ω into $\Omega(x^{\circledast})$ if included already, once ω is marked, the processing of ω is inhibited, thereby preventing the infinite matching of cycles in $\mathcal{O}^*$.

Example 13. Consider the open dictionary $\mathcal{P}^{\circledast}_{[2]}$ in Figure 5 (left) and the symptom pattern $\mathcal{O}^*$ in Figure 6. The extension of $\mathcal{P}^{\circledast}_{[2]}$ based on $\mathcal{O}^*$ by Algorithm 2 is performed as follows (to distinguish from $\mathcal{O}^*$ states, the states of the open dictionary are in bold). Initially, the labeling set $\Omega(\mathbf{0})$ is $\{0\}$, where 0 is the initial state of $\mathcal{O}^*$. Since both *act* and *sby* are matched, the labeling sets of the

Algorithm 2 *Dictionary Extension*

1: **procedure** DICTIONARY EXTENSION($\mathcal{X}^{\circledast}$, $\mathcal{O}^*$)
2: $\mathcal{X}^{\circledast} = (\Sigma, X^{\circledast}, \tau^{\circledast}, x_0^{\circledast})$: an open dictionary of $\mathcal{X}$
3: $\mathcal{O}^* = (\Sigma_\omega, \Omega, \tau_\omega, \omega_0, \Omega_\mathrm{f})$: a symptom pattern of $\mathcal{X}$
4: **begin**
5: Insert ω_0 into the labeling set $\Omega(x_0^{\circledast})$
6: **repeat**
7: Choose a state $x^{\circledast} \in X^{\circledast}$ such that there is an unmarked $\omega \in \Omega(x^{\circledast})$
8: **for all** unmarked $\omega \in \Omega(x^{\circledast})$ **do**
9: **for all** transitions $\langle \omega, o, \omega' \rangle \in \tau_\omega$ **do**
10: **if** $\langle x^{\circledast}, o, x'^{\circledast} \rangle \in \tau^{\circledast}$ **then**
11: Insert ω' into $\Omega(x'^{\circledast})$ **only if** $\omega' \notin \Omega(x'^{\circledast})$
12: **else**
13: $X_o^* \leftarrow \{x'^* \mid x^* \in x^{\circledast}, \langle x^*, o, x'^* \rangle \in \tau(\mathcal{X}^*)\}$
14: $\hat{X}_o^* \leftarrow$ the set of states in $\mathcal{X}^*$ that are reachable from a state in X_o^* by a sequence of transitions $\langle x_i^*, t, x_j^* \rangle$ with t being unobservable
15: $\bar{X}_o^* \leftarrow X_o^* \cup \hat{X}_o^*$
16: **if** $X^{\circledast}$ includes a state $x'^{\circledast} = \bar{X}_o^*$ **then**
17: Insert into $\tau^{\circledast}$ the new transition $\langle x^{\circledast}, o, x'^{\circledast} \rangle$
18: Insert ω' into $\Omega(x'^{\circledast})$ **only if** $\omega' \notin \Omega(x'^{\circledast})$
19: **else**
20: Insert into $X^{\circledast}$ the new state $x'^{\circledast} = \bar{X}_o^*$
21: $\Delta(x'^{\circledast}) \leftarrow \{\delta \mid x^* \in x'^{\circledast}, x^* = (C, L, \delta)\}$
22: Label $x'^{\circledast}$ with the singleton $\Omega(x'^{\circledast}) = \{\omega'\}$
23: Insert into $\tau^{\circledast}$ the new transition $\langle x^{\circledast}, o, x'^{\circledast} \rangle$
24: **end if**
25: **end if**
26: **end for**
27: Mark ω within the labeling set $\Omega(x^{\circledast})$
28: **end for**
29: **until** there is no $x^{\circledast} \in X^{\circledast}$ such that $\Omega(x^{\circledast})$ includes an unmarked state
30: Empty all the nonempty labeling sets $\Omega(x^{\circledast})$
31: **end procedure**

involved states become $\Omega(\mathbf{1}) = \{1\}$ and $\Omega(\mathbf{4}) = \{2\}$. Now, since no transition marked by *opn* exits **4**, the missing dictionary state $x'^{\circledast} = \mathbf{3}$ is generated first computing $X^*_{opn} = \{13\}$, where 13 is the state reached by the $\mathcal{P}^*$ state 9 (cf. Figure 2). However, since no transition exits 13 in $\mathcal{P}^*$, we have $\hat{X}^*_{opn} = \emptyset$ and, hence, $\bar{X}^*_{opn} = \{13\} = \mathbf{3}$. Since it is missing, the state **3** is inserted into $\mathcal{P}^{\circledast}_{[2]}$ and marked by the explanation $\Delta(\mathbf{3}) = \{\{fob, fcs\}\}$. Eventually, the state **3** is labeled with $\Omega(\mathbf{3}) = \{3\}$ and the transition $\langle \mathbf{4}, opn, \mathbf{3} \rangle$ is created. Since no transition exits the state 3 in $\mathcal{O}^*$, the processing of $\omega = 3$ has no effect and the condition of termination in line 29 is true, thereby ending the loop. The updated open dictionary, namely $\mathcal{P}^{\circledast}_{[2,\mathcal{O}]}$, is shown in the center of Figure 5.

Based on Example 13, one may argue that, since the prefix of the symptom $\mathcal{O} = [act, sby, opn]$ up to the second observation, namely $[act, sby]$, is already in

the language of $\mathcal{P}^{\circledast}_{[2]}$, it might be convenient to avoid generating the abduction of $\mathcal{O}$ by *Engine 2*. Instead, the extension of the dictionary might be performed *on the fly* to eventually obtain the explanation from the state **3** created. Actually, this is reasonable in general: Algorithm 2 can actually serve two purposes: either to extend the language of the open dictionary with the language of the symptom pattern or to perform the diagnosis of a given symptom. In either case, Engine 1 matches the observation pattern with the dictionary, whereas Engine 2 performs model-based reasoning to generate the portion of the dictionary that is missing.

7 Scenarios

An open dictionary $\mathcal{X}^{\circledast}$ can be extended with (a possibly infinite number of) new symptoms. The simplest way is adding a symptom $\mathcal{O}$ that was previously explained by *Engine 2*, as in Example 13. If $\mathcal{O}$ is generated in $\mathcal{X}^{\circledast}$ by a path of transitions involving a cycle, then the language of $\mathcal{X}^{\circledast}$ will be extended not only with $\mathcal{O}$, but also with the infinite symptoms involved in the circular path. For example, extending $\mathcal{P}^{\circledast}_{[2]}$ in Figure 5 with the symptom $[act, opn, sby, cls]$ actually extends $\mathcal{P}^{\circledast}_{[2]}$ with the infinite set of symptoms generated by the circular path $0 \to 1 \to 2 \to 5 \to 0$. The dictionary can also be extended based on particular behavioral patterns of the DES, called *scenarios*. A scenario is a behavior of the DES that is considered either most probable or most critical and, hence, is required to be explained efficiently. The idea is to generate the symptom pattern of the scenario and to extend the language of the open dictionary with its language. This way, each symptom generated from now on by a trajectory that conforms with the scenario will be explained by *Engine 1* quickly.

Definition 6. *A* scenario *of a DES $\mathcal{X}$ is a pair $\mathcal{S} = (\Sigma, \mathcal{L})$, where Σ is a subset of the component transitions in $\mathcal{X}$ and $\mathcal{L}$ is a regular language on Σ.*

Since Σ is a subset of the component transitions in $\mathcal{X}$, all the transitions not included in Σ are irrelevant to the scenario. Therefore, in general, a string in $\mathcal{L}$ is not a trajectory of $\mathcal{X}$.

Example 14. The scenario in which the breaker is stuck closed can be defined as $\mathcal{S} = (\Sigma, \mathcal{L})$, where $\Sigma = \{s_3, s_4, b_1, b_2, b_3, b_4\}$ and $\mathcal{L}$ is specified by the regular expression $b_3\, b_3\, b_3^*$ (namely, b_3 repeated at least twice).[3]

Definition 7. *Let $\mathcal{S} = (\Sigma, \mathcal{L})$ be a scenario of $\mathcal{X}$. The* restriction *of a trajectory T in $\mathcal{X}^*$ on Σ is the sequence $T_\Sigma = [t \mid t \in T, t \in \Sigma]$. The* abduction *of $\mathcal{S}$, denoted $\mathcal{X}^*_{\mathcal{S}}$, is a DFA whose language is the set $\{T \mid T \in \mathcal{X}^*, T_\Sigma \in \mathcal{L}\}$.*

In other words, the abduction of a scenario $\mathcal{S}$ is a subspace of $\mathcal{X}^*$ where each trajectory T conforms to one string of the scenario, in the sense that the subsequence of the component transitions in T that are in Σ is a string in $\mathcal{L}$.

[3] A regular expression is defined inductively on the alphabet Σ. The empty symbol ε is a regular expression. If $a \in \Sigma$, then a is a regular expression. If x and y are regular expressions, then the followings are regular expressions: $x \mid y$ (alternative), $x\, y$ (concatenation), $x?$ (optionality), and x^* (repetition zero or more times).

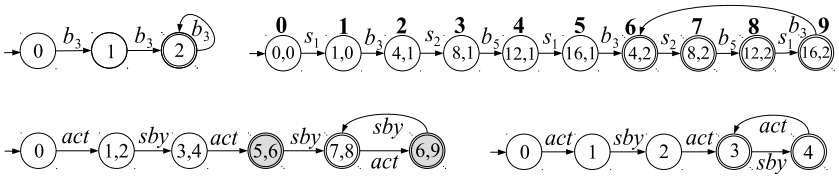


Fig. 7. $\hat{\mathcal{L}}$ (top-left), $\mathcal{P}^*_{\mathcal{S}}$ (top-right), and $\mathcal{O}^*_{\mathcal{S}}$ (bottom).

Example 15. Consider the scenario $\mathcal{S}$ defined in Example 14. The generation of the abduction $\mathcal{P}^*_{\mathcal{S}}$ is based on the DFA recognizing the language $\mathcal{L}$, namely $\hat{\mathcal{L}}$, shown on the top-left of Figure 7. The DFA representing $\mathcal{P}^*_{\mathcal{S}}$ is displayed on the top-right of the same figure, where each state is a pair $(p^*, \hat{\ell})$, where p^* is a state of $\mathcal{P}^*$ and $\hat{\ell}$ a state of $\hat{\mathcal{L}}$. A state $(p^*, \hat{\ell})$ is final when $\hat{\ell}$ is final.

Definition 8. *Let $\mathcal{S} = (\Sigma, \mathcal{L})$ be a scenario of a DES $\mathcal{X}$ and $\mathcal{X}^*_{\mathcal{S}}$ the abduction of $\mathcal{S}$. Let $\mathcal{N}$ be the NFA obtained from $\mathcal{X}^*_{\mathcal{S}}$ by substituting $\langle x, o, x' \rangle$ for every transition $\langle x, t, x' \rangle$, where $(t, o, f) \in \mu(\mathcal{X})$. The* symptom pattern *of the scenario $\mathcal{S}$, denoted $\mathcal{O}^*_{\mathcal{S}}$, is the minimum DFA equivalent to $\mathcal{N}$.*

Example 16. With reference to the abduction $\mathcal{P}^*_{\mathcal{S}}$ determined in Example 15 (top-right of Figure 7), shown on the bottom-left side of Figure 7 is the DFA obtained by determinization of $\mathcal{N}$ (cf. Definition 8), where the states $\{5, 6\}$ and $\{6, 9\}$ are equivalent. The minimal DFA, namely the symptom pattern $\mathcal{O}^*_{\mathcal{S}}$, is shown on the bottom-right side of Figure 7.

The language of the symptom pattern $\mathcal{O}^*_{\mathcal{S}}$ of a scenario $\mathcal{S}$ is composed of all the symptoms with which $\mathcal{S}$ manifests itself to the observer. Still, any such symptom can be implied not only by the trajectories that conform with the scenario, but also by other trajectories. The extension of the open dictionary based on $\mathcal{O}^*_{\mathcal{S}}$ allows for the sound and complete explanation of any symptom in $\mathcal{O}^*_{\mathcal{S}}$.

Example 17. Based on Algorithm 2, extending the open dictionary $\mathcal{P}^{\circledast}_{[2,\mathcal{O}]}$ (center of Figure 5) with the symptom pattern $\mathcal{O}^*_{\mathcal{S}}$ results in the new open dictionary $\mathcal{P}^{\circledast}_{[2,\mathcal{O},\mathcal{S}]}$ shown on the right side of Figure 5.

8 Conclusion

The diagnosis technique presented in this paper is viable and becomes increasingly efficient without requiring the generation of the whole space of the DES; that is, it works while avoiding total knowledge compilation. The open dictionary is assumed to be initialized before the DES is being operated, starting from a prefix of the temporal dictionary, which is then integrated with the symptoms and the candidate diagnoses relevant to a set of scenarios of the DES that are considered worth being diagnosed efficiently. When the DES is being operated, dual knowledge compilation can be applied, in other words, the open dictionary

can be enlarged at any time in two ways, either: (a) by incorporating new compiled knowledge coming from additional scenarios, or (b) by coping with new symptoms explained by *Engine 2*. We are implementing the diagnosis technique presented in this paper in C++. As future research, we plan to extend the technique to other classes of DESs, including complex DESs [10, 17, 18, 16, 14].

References

1. Baroni, P., Lamperti, G., Pogliano, P., Zanella, M.: Diagnosis of large active systems. Artificial Intelligence **110**(1), 135–183 (1999). https://doi.org/10.1016/S0004-3702(99)00019-3
2. Brand, D., Zafiropulo, P.: On communicating finite-state machines. Journal of the ACM **30**(2), 323–342 (1983). https://doi.org/10.1145/322374.322380
3. Cassandras, C., Lafortune, S.: Introduction to Discrete Event Systems, The Kluwer International Series in Discrete Event Dynamic Systems, vol. 11. Kluwer Academic, Boston, MA (1999). https://doi.org/10.1007/978-0-387-68612-7
4. Cimatti, A., Pecheur, C., Cavada, R.: Formal verification of diagnosability via symbolic model checking. In: 18th International Joint Conference on Artificial Intelligence (IJCAI 2003). pp. 363–369 (2003)
5. Console, L., Picardi, C., Ribaudo, M.: Diagnosis and diagnosability using PEPA. In: 14th European Conference on Artificial Intelligence (ECAI 2000). pp. 131–135. IOS Press, Amsterdam (2000)
6. Hamscher, W., Console, L., de Kleer, J. (eds.): Readings in Model-Based Diagnosis. Morgan Kaufmann, San Mateo, CA (1992)
7. Hopcroft, J., Motwani, R., Ullman, J.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading, MA, third edn. (2006)
8. Jiang, S., Huang, Z., Chandra, V., Kumar, R.: A polynomial algorithm for testing diagnosability of discrete event systems. IEEE Transactions on Automatic Control **46**(8), 1318–1321 (2001)
9. Kahneman, D.: Thinking, Fast and Slow. Farrar, Straus and Giroux, New York (2011)
10. Lamperti, G., Quarenghi, G.: Intelligent monitoring of complex discrete-event systems. In: Czarnowski, I., Caballero, A., Howlett, R., Jain, L. (eds.) Intelligent Decision Technologies 2016, Smart Innovation, Systems and Technologies, vol. 56, pp. 215–229. Springer International Publishing Switzerland (2016). https://doi.org/10.1007/978-3-319-39630-9_18
11. Lamperti, G., Zanella, M.: Diagnosis of discrete-event systems from uncertain temporal observations. Artificial Intelligence **137**(1–2), 91–163 (2002). https://doi.org/10.1016/S0004-3702(02)00123-6
12. Lamperti, G., Zanella, M.: A bridged diagnostic method for the monitoring of polymorphic discrete-event systems. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics **34**(5), 2222–2244 (2004)
13. Lamperti, G., Zanella, M.: Monitoring of active systems with stratified uncertain observations. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans **41**(2), 356–369 (2011). https://doi.org/10.1109/TSMCA.2010.2069096
14. Lamperti, G., Zanella, M., Zhao, X.: Abductive diagnosis of complex active systems with compiled knowledge. In: Thielscher, M., Toni, F., Wolter, F. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the 16th International Conference (KR2018). pp. 464–473. AAAI Press, Tempe, Arizona (2018)

15. Lamperti, G., Zanella, M., Zhao, X.: Introduction to Diagnosis of Active Systems. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92733-6
16. Lamperti, G., Zanella, M., Zhao, X.: Knowledge compilation techniques for model-based diagnosis of complex active systems. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) Machine Learning and Knowledge Extraction, Lecture Notes in Computer Science, vol. 11015, pp. 43–64. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99740-7_4
17. Lamperti, G., Zhao, X.: Diagnosis of complex active systems with uncertain temporal observations. In: Buccafurri, F., Holzinger, A., Tjoa, A.M., Weippl, E. (eds.) Availability, Reliability, and Security in Information Systems, Lecture Notes in Computer Science, vol. 9817, pp. 45–62. Springer International Publishing AG Switzerland (2016). https://doi.org/10.1007/978-3-319-45507-5_4
18. Lamperti, G., Zhao, X.: Viable diagnosis of complex active systems. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016). pp. 457–462. Budapest (2016). https://doi.org/10.1109/SMC.2016.7844282
19. Liu, F., Qiu, D.: Diagnosability of fuzzy discrete-event systems: A fuzzy approach. IEEE Transactions on Fuzzy Systems **17**, 372–384 (2009). https://doi.org/10.1109/TFUZZ.2009.2013840
20. Paoli, A., Lafortune, S.: Diagnosability analysis of a class of hierarchical state machines. Journal of Discrete Event Dynamic Systems: Theory and Applications **18**(3), 385–413 (2008)
21. Pencolé, Y.: Diagnosability analysis of distributed discrete event systems. In: 16th European Conference on Artificial Intelligence (ECAI 2004). pp. 43–47. Valencia, Spain (2004)
22. Pencolé, Y., Steinbauer, G., Mühlbacher, C., Travé-Massuyès, L.: Diagnosing discrete event systems using nominal models only. In: 28th International Workshop on Principles of Diagnosis (DX 2017). pp. 169–183. Brescia, Italy (2017)
23. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence **32**(1), 57–95 (1987)
24. Rintanen, J., Grastien, A.: Diagnosability testing with satisfiability algorithms. In: 20th International Joint Conference on Artificial Intelligence (IJCAI 2007). pp. 532–537. Hyderabad, India (2007)
25. Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D.: Diagnosability of discrete-event systems. IEEE Transactions on Automatic Control **40**(9), 1555–1575 (1995)
26. Schumann, A., Huang, J.: A scalable jointree algorithm for diagnosability. In: 23rd National Conference on Artificial Intelligence (AAAI 2008). pp. 535–540. Chicago, IL (2008)
27. Su, X., Zanella, M., Grastien, A.: Diagnosability of discrete-event systems with uncertain observations. In: 25th International Joint Conference on Artificial Intelligence (IJCAI 2016). pp. 1265–1571. New York, NY (2016)
28. Thorsley, D., Teneketzis, D.: Diagnosability of stochastic discrete-event systems. IEEE Transactions on Automatic Control **50**, 476–492 (2005). https://doi.org/10.1109/TAC.2005.844722
29. Ye, L., Dague, P., Yan, Y.: An incremental approach for pattern diagnosability in distributed discrete event systems. In: 21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2012). pp. 123–130. Newark, NJ (2009). https://doi.org/10.1109/ICTAI.2009.75
30. Yoo, T., Lafortune, S.: Polynomial-time verification of diagnosability of partially observed discrete-event systems. IEEE Transactions on Automatic Control **47**(9), 1491–1495 (2002)