Explanatory Diagnosis of Discrete-Event Systems with Temporal Information and Smart Knowledge-Compilation

Nicola Bertoglio[1] , **Gianfranco Lamperti**[1] , **Marina Zanella**[1] , **Xiangfu Zhao**[2]
[1]Department of Information Engineering, University of Brescia, Via Branze 38, Brescia 25123, Italy
[2]School of Computer and Control Engineering, Yantai University, 30, Qingquan RD, Laishan District, Yantai 264005, China
{n.bertoglio001,gianfranco.lamperti,marina.zanella}@unibs.it, xiangfuzhao@gmail.com

Abstract

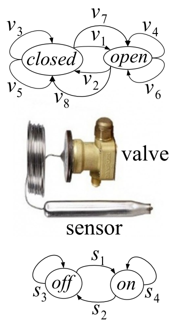
Model-based diagnosis is typically set-oriented. In static systems, such as combinational circuits, a candidate (or diagnosis) is a set of faulty components that explains a set of observations. In discrete-event systems (DESs), a candidate is a set of faulty events occurring in a sequence of state changes that conforms with a sequence of observations. Invariably, a candidate is a set. This set-oriented perspective makes diagnosis of DESs narrow in explainability, owing to the lack of any temporal knowledge relevant to the faults within a candidate, along with the inability to discriminate between single and multiple occurrences of the same fault. Embedding temporal knowledge in a candidate, such as the relative temporal ordering of faults and the multiplicity of the same fault, may be essential for critical decision making. To favor explainability, the notions of temporal fault, explanation, and explainer are introduced in diagnosis of DESs. The explanation engine reacts to a given sequence of observations by generating and refining in real-time a sequence of regular expressions, where the language of each expression is a set of temporal faults. Moreover, to avoid total knowledge compilation, the explainer can be generated incrementally either offline, based on meaningful behavioral scenarios, or online, when being operated in solving specific diagnosis problems.

1 Introduction

Model-based diagnosis (Reiter 1987; Hamscher, Console, and de Kleer 1992) exploits the model of a system in order to find the causes of its abnormal behavior, based on some observations. For diagnosing a dynamical system (Struss 1997), a discrete-event system (DES) model can be adopted (Cassandras and Lafortune 2008), this being either a Petri net (Basile 2014; Cong et al. 2018) or a net of communicating finite automata (Brand and Zafiropulo 1983), an automaton for each component, like in the current paper. Although an automaton possibly represents just the nominal behavior of a DES component (Pencolé et al. 2017), usually each state transition is either *normal* or *abnormal*, as in the seminal work by (Sampath et al. 1995; Sampath et al. 1996). The input of the diagnosis task is a sequence of temporally ordered observations, called a *temporal observation*. The output is a set of *candidates*, with each candidate being a set of *faults*, where a fault is associated with an abnormal state transition. Diagnosing a DES becomes a form of *abductive* reasoning (McIlraith 1998), inasmuch the candidates are generated based on the trajectories (sequences of state transitions) of the DES that entail the temporal observation. The approach in (Sampath et al. 1995) performs the abduction offline, thus compiling the DES models into a *diagnoser*, a data structure that is consulted online in order to produce a new set of candidates upon perceiving each observation (*monitoring-based diagnosis*). In the *active-system* approach (Lamperti, Zanella, and Zhao 2018a), instead, the abduction is performed online, a possibly costly operation that, however, being driven by the temporal observation, can focus on the trajectories that imply this sequence only. The diagnosis output is the set of candidates relevant to the (possibly infinite) set of trajectories of the DES that produce the temporal observation. Since the domain of faults is finite, both the candidates and the diagnosis output are finite and bounded. Still, in both the diagnoser approach and the active-system approach, a candidate is a *set* of faults. Consequently, the diagnosis output is devoid of any temporal feature while in the real world faults occur in a specific temporal order. One may argue that, since a new set of candidates is output upon the reception of a new observation, it is possible to ascertain whether some additional faults have occurred with respect to the previous observation. However, one cannot ascertain whether a fault occurred previously has occurred again, in other words, no information about *intermittent* faults is provided.

In a perspective of explainable diagnosis and, more generally, of explainable AI, in this paper, a candidate is not a set of faults, instead it is a *temporal fault*, namely the (possibly unbounded) sequence of faults relevant to a trajectory that produces the temporal observation. Therefore, the diagnosis output is the (possibly infinite) set of temporal faults relevant to the (possibly infinite) set of trajectories of the DES that imply the temporal observation (received so far). Since the diagnosis output is a regular language over the alphabet of faults, it can be represented by a regular expression. The temporal knowledge embedded in temporal faults can support the diagnostician in making critical decisions to perform suitable repair actions. This novel characterization of the diagnosis output is the first contribution of the paper.

The second one is a theoretical framework for defining the task that generates such an output. A data structure compiled offline, called *explainer*, which enables the online efficient computation of the new diagnosis output is proposed. The



Component transition	Description
$v_1 = \langle closed, (op, \emptyset), open \rangle$	The valve reacts to the open event by opening
$v_2 = \langle open, (cl, \emptyset), closed \rangle$	The valve reacts to the close event by closing
$v_3 = \langle closed, (op, \emptyset), closed \rangle$	The valve does not react to the open event and remains closed
$v_4 = \langle open, (cl, \emptyset), open \rangle$	The valve does not react to the close event and remains open
$v_5 = \langle closed, (cl, \emptyset), closed \rangle$	The valve reacts to the close event by remaining closed
$v_6 = \langle open, (op, \emptyset), open \rangle$	The valve reacts to the open event by remaining open
$v_7 = \langle closed, (cl, \emptyset), open \rangle$	The valve reacts to the close event by opening
$v_8 = \langle open, (op, \emptyset), closed \rangle$	The valve reacts to the open event by closing
$s_1 = \langle off, (high, \{op\}), on \rangle$	The sensor detects high temperature and generates the open event
$s_2 = \langle on, (low, \{cl\}), off \rangle$	The sensor detects low temperature and generates the close event
$s_3 = \langle off, (high, \{cl\}, off \rangle$	The sensor detects high temperature, yet generates the close event
$s_4 = \langle on, (low, \{op\}), on \rangle$	The sensor detects low temperature, yet generates the open event

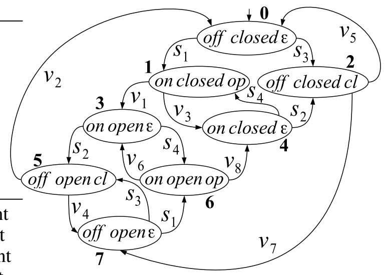


Figure 1: DES $\mathcal{P}$ (left), details of component transitions (center), and behavioral space $\mathcal{P}^*$ (right).

explainer acts somewhat as a counterpoint to Sampath's diagnoser in set-oriented diagnosis of DESs.

The third contribution of the paper is an algorithm, called *Explanation Engine*, which, every time a new observation has been perceived, is able not only to generate an updated set of candidates but also to prune the previously output candidates that are no longer consistent with the temporal observation. This progressive refinement of results is key to enhancing the explainability of monitoring-based diagnosis.

Since the generation of the explainer requires total knowledge compilation, which turns out to be impractical in real applications because of the exponential explosion of the number of states, a fourth contribution is a technique for building a partial explainer upfront, which can be subsequently extended based on either meaningful behavioral scenarios of the DES or new diagnosis problems that are being solved over time by a *Smart Explanation Engine*.

2 DES Modeling

A DES $\mathcal{X}$ is a network of *components*, where the behavior of each component is modeled as a communicating automaton. A component is endowed with input and output *pins*, where each output pin is connected with an input pin of another component by a *link*.

A component transition can be triggered either by an external event coming from the outside of $\mathcal{X}$ or by an internal event coming from another component in $\mathcal{X}$. When an event occurs, a component of the DES may react by performing a transition. When performing a transition, a component consumes the triggering (input) event and possibly generates new events on its output pins, which are bound to trigger the transitions of other components. A transition generating an event on an output pin can occur only if this pin is not occupied by another event.

Assuming that only one component transition at a time can occur, the process that moves a DES from the initial state can be represented by a sequence of component transitions, called a *trajectory* of $\mathcal{X}$. A contiguous subsequence of a trajectory is a *trajectory segment*. At the occurrence of a component transition, $\mathcal{X}$ changes state, with a state x of $\mathcal{X}$ being a pair (C, L), where C is the array of the current component states and L the array of the (possibly empty) current events placed in links. The (possibly infinite) set of trajectories of $\mathcal{X}$ is specified by a deterministic finite automaton (DFA), the (*behavioral*) *space* of $\mathcal{X}$,

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

where Σ (the alphabet) is the set of component transitions, X is the set of states, τ is the deterministic transition function mapping a state and a component transition into a new state, $\tau : X \times \Sigma \mapsto X$, and x_0 is the initial state. Each trajectory T is a string of the language of $\mathcal{X}^*$, hence we write $T \in \mathcal{X}^*$.

For diagnosis purposes, the model of $\mathcal{X}$ needs to be enriched by a *mapping table*. 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* μ 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*; likewise, if $f \neq \varepsilon$, then t is *faulty*, else t is *normal*. Based on $\mu(\mathcal{X})$, each trajectory T in $\mathcal{X}^*$ can be associated with a *temporal observation*, this being the sequence of observations involved in T, namely

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

The set of temporal observations relevant to all the trajectories in $\mathcal{X}^*$ is the *observation language* of $\mathcal{X}^*$, namely

$$OBS(\mathcal{X}^*) = \{ Obs(T) \mid T \in \mathcal{X}^* \}. \tag{3}$$

In the literature, a trajectory T is also associated with a diagnosis, namely the set of faults involved in T. As such, a diagnosis does not indicate the temporal relationships between faults, nor does it account for multiple occurrences of the same fault. On the other hand, treating a diagnosis as a set of faults guarantees that the domain of possible diagnoses is finite, being bounded by the powerset of the domain of faults. In contrast with this classical perspective, we introduce the notion of a temporal fault.

Definition 1 (temporal fault)**.** *Let T be a trajectory of a DES. The* temporal fault *of T is the sequence of faults*

$$Flt(T) = [\, f \mid t \in T, (t, o, f) \in \mu(\mathcal{X}), f \neq \varepsilon \,]. \tag{4}$$

t	o	f
s_1	*sen*	ε
s_2	*sen*	ε
s_3	ε	α_1
s_4	ε	α_2
v_1	*val*	ε
v_2	*val*	ε
v_3	ε	β_1
v_4	ε	β_2
v_5	*val*	ε
v_6	*val*	ε
v_7	*val*	β_3
v_8	*val*	β_4

o	*Observation description*
sen	The sensor s performs an action
val	The valve v performs an action

f	*Fault description*
α_1	s sends the *cl* command instead of *op*
α_2	s sends the *op* command instead of *cl*
β_1	v remains closed on the *op* command
β_2	v remains open on the *cl* command
β_3	v opens on the *cl* command
β_4	v closes on the *op* command

Figure 2: Mapping table $\mu(\mathcal{P})$ (left) and description (right).

Since the length of T is in general unbounded, the length of both $Obs(T)$ and $Flt(T)$ is in general unbounded. A contiguous subsequence of a temporal fault is called a *temporal-fault segment*.

Example 1 (DES)**.** Centered on the left side of Fig. 1 is a DES called $\mathcal{P}$ (protection), which includes two components, a (temperature) sensor s and a valve v, and one link connecting the (single) output pin of s with the (single) input pin of v. The protection is designed to control a cooling system. When the temperature becomes high, the sensor commands the valve to open in order to let the cooling fluid to flow. Vice versa, when the temperature returns to normal, the sensor commands the valve to close. The model of s, outlined under $\mathcal{P}$, involves two states (denoted by circles) and four transitions (denoted by arcs). The model of v is shown above $\mathcal{P}$. 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 the table within Fig. 1. The space of $\mathcal{P}$, namely $\mathcal{P}^*$, is depicted on the right side of Fig. 1, where each state is identified by a triple $(\bar{s}, \bar{v}, e)$, with $\bar{s}$ being the state of the sensor, $\bar{v}$ the state of the valve, and e the internal event in the link (ε means no event). For simple referencing, the states of $\mathcal{P}^*$ are renamed $0 \cdots 7$, where 0 is the initial state. Owing to cycles, the set of possible trajectories of $\mathcal{P}$ is infinite, one of them being $T = [s_3, v_5, s_1, v_3, s_4, v_3, s_2, v_5]$. The mapping table $\mu(\mathcal{P})$ is shown on the left side of Fig. 2, with observations and faults being described on the right side. Only one observation is provided for both the sensor and the valve, namely *sen* and *val*, respectively, each being associated with several (still not all) transitions. Based on $\mu(\mathcal{P})$ and the trajectory T defined above, we have $Obs(T) = [val, sen, sen, val]$ and $Flt(T) = [\alpha_1, \beta_1, \alpha_2, \beta_1]$. In a set-oriented perspective, the diagnosis of T would be the set $\{\alpha_1, \alpha_2, \beta_1\}$, where neither the temporal ordering of faults nor the double occurrence of β_1 is contemplated.

3 Explanation and Explainability

The essential goal in diagnosing a DES is generating the set of candidates relevant to a temporal observation $\mathcal{O}$. In this paper, in contrast with the classical set-oriented perspective, a candidate is a temporal fault that is produced by a trajectory that entails $\mathcal{O}$. The (possibly infinite) set of candidates is the *explanation* of $\mathcal{O}$, as formalized in Def. 2.

Definition 2 (explanation)**.** *Let $\mathcal{O} = [o_1, \ldots, o_n]$ be a temporal observation of $\mathcal{X}$ and T a trajectory of $\mathcal{X}$ such that $Obs(T) = \mathcal{O}$. Let $T_{[i]}$, $i \in [0 \mathinner{..} n]$, 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* explanation *of $\mathcal{O}$ is a sequence*

$$\mathcal{E}(\mathcal{O}) = [\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_n] \quad (5)$$

where each $\mathcal{F}_i$, $i \in [0 \mathinner{..} n]$, is the minimal (possibly unbounded) set of temporal faults defined as follows:

$$T \in \mathcal{X}^*, Obs(T) = \mathcal{O}, \forall i \in [0 \mathinner{..} n] \left(\mathcal{F}_i \supseteq \left\{ Flt\left(T_{[i]}\right) \right\}\right). \quad (6)$$

Example 2 (explanation)**.** Let $\mathcal{O} = [val, sen, sen]$ be a temporal observation of the DES $\mathcal{P}$ defined in Example 1. Based on the space $\mathcal{P}^*$ in Fig. 1 and the mapping table $\mu(\mathcal{P})$ in Fig. 2, the language of the trajectories generating $\mathcal{O}$ can be represented by a regular expression[1], namely $s_3 v_5 s_1 v_3 (s_4 v_3)^* s_2$. Hence, according to Def. 2,

$$\mathcal{E}(\mathcal{O}) = [\alpha_1, \alpha_1, \alpha_1 \beta_1 (\alpha_2 \beta_1)^*, \alpha_1 \beta_1 (\alpha_2 \beta_1)^*].$$

Each of the four sets of temporal faults in $\mathcal{E}(\mathcal{O})$ is represented by a regular expression. In a classical set-oriented perspective, the set of candidates corresponding to the fourth regular expression (which equals the third one) would be $\{\{\alpha_1, \beta_1\}, \{\alpha_1, \alpha_2, \beta_1\}\}$, in which case the diagnostician knows that both fault α_1 and β_1 have certainly occurred, whereas the occurrence of fault α_2 is uncertain; still, he has no hint about how frequently such faults have manifested themselves and in which temporal order. If, instead, the regular expression is given, the diagnostician knows not only that both faults α_1 and β_1 have certainly occurred, but that α_1 has occurred just once and it was the first, while β_1 was the second. In addition, he knows that, if α_2 has occurred, it has occurred after them, and it may have occurred several times, every time being followed by β_1. These details may be essential to explain what has happened inside the system in order to make critical decisions for suitable recovery actions.

It can be proven that the sets of temporal faults in eqn. (5) are invariably regular languages (like in Example 2). This allows a (possibly infinite) set of temporal faults to be always represented by a (finite) regular expression.

Depending on the degree of observability, a set of temporal faults in an explanation can be either finite or infinite. In this respect, the notion of *explainability* formalized below provides a classification of DESs.

[1]A regular expression is defined inductively over an alphabet Σ as follows. The empty symbol ε and every $a \in \Sigma$ is a regular expression. If x and y are regular expressions, then the followings are regular expressions: (x) (parentheses may be used), $x \mid y$ (alternative), xy (concatenation), $x?$ (optionality), x^* (repetition zero or more times), and x^+ (repetition one or more times). When parentheses are missing, the concatenation has precedence over the alternative, while optionality and repetition have the highest precedence; for example, $ab^* \mid cd?$ denotes $(a(b^*)) \mid (c(d?))$ (Hopcroft, Motwani, and Ullman 2006).

Definition 3 (explainability)**.** *Let $\mathcal{O} = [o_1, \ldots, o_n]$ be a generic temporal observation of a DES $\mathcal{X}$ and*

$$\mathcal{E}(\mathcal{O}) = [\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_n]$$

the explanation of $\mathcal{O}$. If, for each $\mathcal{O}$ and for each language $\mathcal{F}_i$, $i < n$, $\mathcal{F}_i$ is finite, then $\mathcal{X}$ is enumerably explainable.[2] *If $\mathcal{X}$ is enumerably explainable and, more specifically, for each $\mathcal{O}$ and for each language $\mathcal{F}_i$, $i < n$, $\mathcal{F}_i$ is a singleton, then $\mathcal{X}$ is* strongly explainable. *If $\mathcal{X}$ is not enumerably explainable (and hence, even more so, not strongly explainable), then $\mathcal{X}$ is* weakly explainable.

From Def. 3, we can infer that a DES is weakly explainable in case its (behavioral) space contains unobservable cycles that include some faulty transition(s). This feature of the space can give rise to highly uncertain online situations: for some given temporal observations, it is impossible to know whether some faults have occurred and how many times they have occurred. As far as enumerably explainable DESs are concerned, the number of temporal faults relevant to each trajectory that entails a given temporal observation up to the last but one perceived observation is finite. The number of trajectories that entail the whole temporal observation perceived up to now may instead be unbounded, hence the latest set of temporal faults of the explanation is unbounded too. Once a new observation has been perceived, however, this set becomes finite.

The reason for the enumerable explainability not enforcing any enumerability constraint on the latest set of temporal faults in the explanation is that this set may be unbound because it includes not only the trajectories whose last transition is the one generating the latest received observation but also all the trajectories where this transition is followed by unobservable ones, several of which may be inconsistent with the next observation. Enumerable explainability does not guarantee that all the sets of temporal faults but the latest one are uncertainty-free; in fact, every set that includes more than one temporal fault is uncertain as we do not know the sequence of faults actually occurred. Hence, only strong explainability guarantees full knowledge of the faults occurred up to the last but one perceived observation.

Example 3 (explainability)**.** Based on $\mathcal{E}(\mathcal{O})$ in Example 2, we conclude that $\mathcal{P}$ is weakly explainable, as the regular language denoted by $\mathcal{F}_2 = \alpha_1\beta_1(\alpha_2\beta_1)^*$ is infinite owing to the unbounded repetition $(\alpha_2\beta_1)^*$.

In order to support the efficient generation of the explanation of any temporal observation, a data structure called an *explainer* is introduced in the next section.

4 Total Knowledge Compilation

The explainer of a DES $\mathcal{X}$ is a nondeterministic finite automaton (NFA) resulting from the total compilation of $\mathcal{X}$. Roughly, it is a transformation of the space $\mathcal{X}^*$ extended with explicit information on faults. The alphabet of the explainer is a set of triples $(o, \mathcal{L}, f)$, where o is an observation of $\mathcal{X}$, $\mathcal{L}$ is a regular language on the faults of $\mathcal{X}$, and f is a (possibly empty) fault. Each state of the explainer (namely a *fault space*) embodies a sort of local explanation defined by regular languages (in fact, regular expressions) on faults.

[2]No constraint is required for $\mathcal{F}_n$; still, when $\mathcal{O}$ is extended to $\mathcal{O}'$ by o_{n+1}, $\mathcal{F}_n$ is expected to become finite in $\mathcal{O}'$.

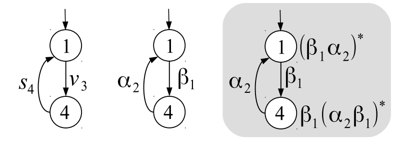


Figure 3: Genesis of the fault space $\mathcal{P}_1^*$ (shaded on the right).

Definition 4 (fault space)**.** *Let $\mathcal{X}^*$ be the space of a DES $\mathcal{X}$ having mapping table $\mu(\mathcal{X})$, $\mathbf{F}$ the set of faults of $\mathcal{X}$, and $\bar{x}$ a state in $\mathcal{X}^*$. The* fault space *of $\bar{x}$ is an NFA*

$$\mathcal{X}^*_{\bar{x}} = (\Sigma, X, \tau, x_0) \tag{7}$$

where $\Sigma = \mathbf{F} \cup \{\varepsilon\}$ is the alphabet, X is the subset of the states of $\mathcal{X}^*$ that are reachable from $\bar{x}$ by unobservable transitions, $x_0 = \bar{x}$ is the initial state, and $\tau : X \times \Sigma \mapsto 2^X$ is the transition function, where $\langle x_1, f, x_2\rangle$ is an arc in τ iff $\langle x_1, t, x_2\rangle$ is a transition in $\mathcal{X}^*$ and $(t, o, f) \in \mu(\mathcal{X})$, with $o = \varepsilon$. Each state $x \in X$ is marked with the regular language of the temporal-fault segments of the trajectory segments in $\mathcal{X}^*$ from $\bar{x}$ to x, denoted $\mathcal{L}(x)$.

Example 4 (fault space)**.** With reference to the DES $\mathcal{P}$ introduced in Example 1, shown in Fig. 3 is the genesis of the fault space $\mathcal{P}_1^*$ based on Def. 4. The graph on the left represents the portion of the space $\mathcal{P}^*$ that is reached by unobservable transitions starting from the state 1. Then, the identifiers of the transitions are replaced with the corresponding faults, thereby obtaining the graph in the center. The actual fault space is depicted on the right of Fig. 3, where states 1 and 4 are marked with regular expressions on faults.

The decoration of the states in the fault space in Fig. 3 can be carried out by inspection of the NFA shown in the center of the figure. What we need, however, is a general technique allowing for the automatic decoration of the states within a fault space. To this end, we have exploited and adapted the algorithm proposed in (Brzozowski and McCluskey 1963) in the context of sequential circuit state diagrams. Essentially, this algorithm takes as input an NFA and generates the regular expression of the language accepted by this NFA. However, in a fault space, all states need to be marked with the relevant regular expressions. Thus, we have extended the algorithm to decorate all the states in one processing of the NFA (rather than one processing for each state).

Definition 5 (explainer)**.** *Let $\mathcal{X}^* = (\Sigma, X, \tau, x_0)$ be the space of $\mathcal{X}$, $\mathbf{O}$ the set of observations of $\mathcal{X}$, $\mathbf{F}$ the set of faults of $\mathcal{X}$, and $\mathbf{L}$ the set of regular languages on $\mathbf{F} \cup \{\varepsilon\}$. The* explainer *of $\mathcal{X}$ is an NFA*

$$\mathcal{X}^{\mathcal{E}} = (\Sigma', X', \tau', x'_0) \tag{8}$$

where $\Sigma' \subseteq \mathbf{O} \times \mathbf{L} \times (\mathbf{F} \cup \{\varepsilon\})$ is the alphabet, X' is the set of states, where each state is a fault space of a specific state of $\mathcal{X}^*$, $x'_0 = \mathcal{X}^*_{x_0}$ is the initial state, and τ' is the (nondeterministic) transition function,

$$\tau' : (X' \times X) \times \Sigma' \mapsto 2^{(X' \times X)},$$

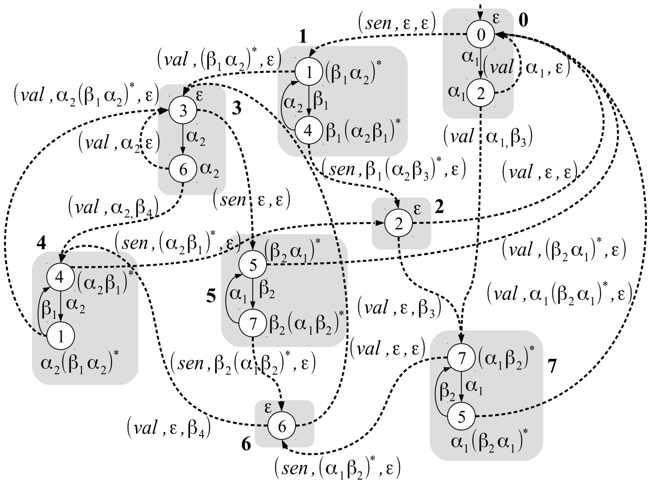

$(sen, \varepsilon, \varepsilon)$
ε 0
1
$(val, (\beta_1\alpha_2)^*, \varepsilon)$
$(\beta_1\alpha_2)^*$
$(val, \alpha_1, \varepsilon)$
$(val, \alpha_2(\beta_1\alpha_2)^*, \varepsilon)$
3
$\beta_1(\alpha_2\beta_1)^*$
$(val, \alpha_2, \varepsilon)$
$(val\ \alpha_1, \beta_3)$
$(sen, \beta_1(\alpha_2\beta_3)^*, \varepsilon)$
$(val, \varepsilon, \varepsilon)$
(val, α_2, β_4)
$(sen, \varepsilon, \varepsilon)$
ε 2
4
$(sen, (\alpha_2\beta_1)^*, \varepsilon)$
$(\beta_2\alpha_1)^*$
$(val, (\beta_2\alpha_1)^*, \varepsilon)$
$(\alpha_2\beta_1)^*$
5
$(val, \alpha_1(\beta_2\alpha_1)^*, \varepsilon)$
$(val, \varepsilon, \beta_3)$
$\beta_2(\alpha_1\beta_2)^*$
$\alpha_2(\beta_1\alpha_2)^*$
$(val, \varepsilon, \varepsilon)$
$(\alpha_1\beta_2)^*$
7
$(sen, \beta_2(\alpha_1\beta_2)^*, \varepsilon)$
$(val, \varepsilon, \beta_4)$
6
$\alpha_1(\beta_2\alpha_1)^*$
$(sen, (\alpha_1\beta_2)^*, \varepsilon)$

Figure 4: Explainer of the DES $\mathcal{P}$, namely $\mathcal{P}^{\mathcal{E}}$.

where $\langle (x'_1, x_1), (o, \mathcal{L}(x_1), f), (x'_2, x_2)\rangle$ *is an arc in* τ' *iff* x_1 *is a state in* x'_1, $\langle x_1, t, x_2\rangle \in \tau$, $(t, o, f) \in \mu(\mathcal{X})$, $o \neq \varepsilon$, *and* $x'_2 = \mathcal{X}^*_{x_2}$.

Example 5 (explainer)**.** With reference to the DES $\mathcal{P}$, shown in Fig. 4 is the explainer $\mathcal{P}^{\mathcal{E}}$, where the states (fault spaces) are renamed $0 \cdots 7$. Unlike component transitions within fault spaces, which are represented with plain arcs, the transitions between states of $\mathcal{P}^{\mathcal{E}}$ are depicted as dashed arcs that are marked with the relevant triples.

A trajectory in $\mathcal{X}^{\mathcal{E}}$ is the sequence of triples $(o, \mathcal{L}, f)$ marking a contiguous sequence of transitions in $\mathcal{X}^{\mathcal{E}}$ starting from the initial state. Likewise, the temporal observation of a trajectory T in $\mathcal{X}^{\mathcal{E}}$ is defined as

$$Obs(T) = [\, o \mid (o, \mathcal{L}, f) \in T\,]. \tag{9}$$

The notion of the observation language of $\mathcal{X}$ defined in eqn. (3) can be naturally extended to $\mathcal{X}^{\mathcal{E}}$ as follows:

$$OBS\left(\mathcal{X}^{\mathcal{E}}\right) = \left\{ Obs(T) \mid T \in \mathcal{X}^{\mathcal{E}} \right\}. \tag{10}$$

Proposition 1. *The observation language of the explainer of* $\mathcal{X}$ *equals the observation language of the space of* $\mathcal{X}$,

$$OBS\left(\mathcal{X}^{\mathcal{E}}\right) = OBS(\mathcal{X}^*). \tag{11}$$

This can be proven easily by considering that each sequence of observations generated by a trajectory in $\mathcal{X}^{\mathcal{E}}$ can be generated by a trajectory in $\mathcal{X}^*$ and vice versa. In fact, each state in $\mathcal{X}^{\mathcal{E}}$ is a fault space, namely a subspace of $\mathcal{X}^*$ involving unobservable trajectory segments.

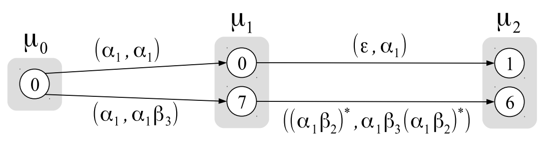

μ_0
μ_1
μ_2
(α_1, α_1)
(ε, α_1)
$(\alpha_1, \alpha_1\beta_3)$
$((\alpha_1\beta_2)^*, \alpha_1\beta_3(\alpha_1\beta_2)^*)$
0
0
7
1
6

Figure 5: Monitoring trace $\mathcal{M}([val, sen])$ for the DES $\mathcal{P}$.

5 Monitoring and Explanation Trace

Given an explainer $\mathcal{X}^{\mathcal{E}}$, the explanation of a temporal observation $\mathcal{O}$ is generated by tracing $\mathcal{O}$ on $\mathcal{X}^{\mathcal{E}}$, thereby yielding a *monitoring trace*, as formalized below.

Definition 6 (monitoring trace)**.** *Let* $\mathcal{O} = [o_1, \dots, o_n]$ *be a temporal observation of* $\mathcal{X}$ *and* $\mathcal{X}^{\mathcal{E}} = (\Sigma, X, \tau, x_0)$ *the explainer of* $\mathcal{X}$. *The* monitoring trace *of* $\mathcal{O}$ *is a directed graph*

$$\mathcal{M}(\mathcal{O}) = (M, A, \mu_0) \tag{12}$$

where $M = \{\mu_0, \mu_1, \dots, \mu_n\}$ *is the set of nodes,* A *is the set of arcs, and* μ_0 *is the initial node. Each node* $\mu_i \in M$, $i \in [0 \mathinner{..} n]$, *is a subset of* X, *with* $\mu_0 = \{x_0\}$. *Each* $\mu_i \neq \mu_0$ *contains the states of* $\mathcal{X}^{\mathcal{E}}$ *that are reached in* $\mathcal{X}^{\mathcal{E}}$ *by the states in* μ_{i-1} *via a transition marked with a triple where the observation is* o_i. *Each arc exiting a state* $x \in \mu_i$ *is marked with a pair* $(\mathcal{L}, \mathcal{L}')$, *where* $\mathcal{L}$ *and* $\mathcal{L}'$ *are languages of temporal-fault segments. There is an arc from a state* $x \in \mu_i$ *to a state* $x' \in \mu_{i+1}$, $i \in [0 \mathinner{..} (n-1)]$, *marked with* $(\mathcal{L}, \mathcal{L}')$, *iff there is a transition in* $\mathcal{X}^{\mathcal{E}}$ *from a space state in* x *to a space state in* x' *that is marked with* $(o_{i+1}, \mathcal{L}, f)$, *whereas* $\mathcal{L}'$ *is defined as follows. Let* $\mathcal{R}$ *be either* ε, *when* $i = 0$, *or* $(\mathcal{L}'_1 | \mathcal{L}'_2 | \dots | \mathcal{L}'_k)$, *when* $i \neq 0$, *where* $(\mathcal{L}_j, \mathcal{L}'_j)$ *is the pair marking the* j*-th arc entering* $x \in \mu_i$, $j \in [1 \mathinner{..} k]$. *Then,* $\mathcal{L}' = \mathcal{R}\mathcal{L}f$.

Example 6 (monitoring trace)**.** Let $\mathcal{O} = [val, sen]$ be a temporal observation of the DES $\mathcal{P}$. The corresponding monitoring trace $\mathcal{M}(\mathcal{O})$ is outlined in Fig. 5.

The monitoring trace of $\mathcal{O}$ allows for the distillation of an *explanation trace* (formalized below), which turns out to equal the explanation of $\mathcal{O}$ (Proposition 2).

Definition 7 (explanation trace)**.** *Let* $\mathcal{M}(\mathcal{O})$ *be a monitoring trace with nodes* $\mu_0, \mu_1, \dots, \mu_n$. *The* explanation trace *of* $\mathcal{M}(\mathcal{O})$ *is a sequence*

$$\mathcal{E}(\mathcal{M}(\mathcal{O})) = [\mathcal{L}(\mu_0), \mathcal{L}(\mu_1), \dots, \mathcal{L}(\mu_n)] \tag{13}$$

where $\mathcal{L}(\mu_i)$, $i \in [0 \mathinner{..} n]$, *is a language of temporal faults defined as follows. Let* $\mu_i = \{x_1, \dots, x_m\}$. *Let* $x_j \in \mu_i$. *Let* $\mathcal{L}_{\text{in}}(x_j)$ *be either* ε, *when* $i = 0$, *or the alternative* $(\mathcal{L}_1 | \mathcal{L}_2 | \dots | \mathcal{L}_m)$, *when* $i \neq 0$, *where* $(\mathcal{L}'_k, \mathcal{L}_k)$, $k \in [1 \mathinner{..} m]$, *is the pair marking the* k*-th arc entering* x_j. *Let* $\mathcal{L}_{\text{out}}(x_j)$ *be either* $(\mathcal{L}'_1 | \mathcal{L}'_2 | \dots | \mathcal{L}'_p)$, *when* $i \neq n$, *where* $(\mathcal{L}'_h, \mathcal{L}_h)$, $h \in [1 \mathinner{..} p]$, *is the pair marking the* h*-th arc exiting* x_j, *or* $(\mathcal{L}(x_{j_1}) | \mathcal{L}(x_{j_2}) | \dots | \mathcal{L}(x_{j_r}))$, *when* $i = n$, *where* $x_{j_1}, \dots, x_{j_r}$ *are the space states in* x_j *and* $\mathcal{L}(x_{j_v})$, $v \in [1 \mathinner{..} j_r]$, *is the language marking* x_{j_v}. *Then,* $\mathcal{L}(\mu_i)$ *is the alternative of pairwise concatenated languages*

$$\mathcal{L}(\mu_i) = (\mathcal{L}_{\text{in}}(x_1)\mathcal{L}_{\text{out}}(x_1) \mid \dots \mid \mathcal{L}_{\text{in}}(x_m)\mathcal{L}_{\text{out}}(x_m)).$$

Example 7 (explanation trace)**.** With reference to the monitoring trace $\mathcal{M}(\mathcal{O})$ displayed in Fig. 5, with $\mathcal{O} = [val, sen]$, the corresponding explanation trace is

$$\mathcal{E}(\mathcal{M}(\mathcal{O})) = [\alpha_1, (\alpha_1 | \alpha_1\beta_3(\alpha_1\beta_2)^*), (\alpha_1((\beta_1\alpha_2)^* | \beta_1(\alpha_2\beta_1)^*) | \alpha_1\beta_3(\alpha_1\beta_2)^*)]$$

which, by factorization of α_1, can be rewritten as

$$\mathcal{E}(\mathcal{M}(\mathcal{O})) = [\alpha_1, \alpha_1(\beta_3(\alpha_1\beta_2)^*)?, \alpha_1(((\beta_1\alpha_2)^* | \beta_1(\alpha_2\beta_1)^*) | \beta_3(\alpha_1\beta_2)^*)].$$

Proposition 2. *Let $\mathcal{M}(\mathcal{O})$ be a monitoring trace. The explanation trace of $\mathcal{M}(\mathcal{O})$ equals the explanation of $\mathcal{O}$,*

$$\mathcal{E}(\mathcal{M}(\mathcal{O})) = \mathcal{E}(\mathcal{O}). \tag{14}$$

Proof (*sketch*). Let $\mathcal{O} = [o_1, \ldots, o_n]$ be a temporal observation of a DES $\mathcal{X}$. According to Def. 2, the explanation $\mathcal{E}(\mathcal{O})$ is a sequence $[\mathcal{F}_0, \ldots, \mathcal{F}_n]$, where each $\mathcal{F}_i$, $i \in [0 \mathinner{..} n]$, is a set of temporal faults generated by considering all the trajectories T of $\mathcal{X}$ such that $Obs(T) = \mathcal{O}$. $\mathcal{F}_i$ is composed of the temporal faults $Flt(T_{[i]})$, where, if $i < n$, $T_{[i]}$ is the prefix of T up to the transition preceding the $(i+1)$-th observable transition; otherwise, when $i = n$, $T_{[i]} = T$.

There is an isomorphism between the trajectories $T_{[i]}$ and the paths in $\mathcal{M}(\mathcal{O})$, including the transitions within the nodes of $\mathcal{M}(\mathcal{O})$, from the initial state of μ_0 to a state in μ_i that is either exited by an arc, when $i < n$, or any state in μ_i, when $i = n$. In fact, each node μ_i of $\mathcal{M}(\mathcal{O})$ is a fault space (Def. 4), which involves the portion of $\mathcal{X}^*$ that is reachable from the initial state of μ_i by unobservable transitions, with such transitions being marked with the corresponding faults.

Let $(\mathcal{L}, \mathcal{L}')$ be the pair marking an arc in $\mathcal{M}(\mathcal{O})$ entering the initial state $\bar{x}$ of a state in μ_i. Then, $\mathcal{L}'$ is the language of the temporal faults relevant to any trajectory of $\mathcal{X}$ ending in $\bar{x}$. This can be proven by induction. According to Def. 6, if the arc considered exits μ_0, then $\mathcal{L}' = \mathcal{L}f$, where $\mathcal{L}$ is the language of temporal faults of the unobservable trajectories up to the state exited by the arc and f is the (possibly empty fault) associated with the observable transition corresponding to the arc. Hence, $\mathcal{L}f$ is the language of the temporal faults relevant to the trajectories up to the transition corresponding to the arc. If this property holds for an arc entering a state in μ_i, $i \in [1 \mathinner{..} (n-1)]$, then it holds for a successive arc entering a state in μ_{i+1}. In fact, based on Def. 6, $\mathcal{L}' = \mathcal{R}\mathcal{L}f$, where $\mathcal{R} = (\mathcal{L}'_1 | \mathcal{L}'_2 | \ldots | \mathcal{L}'_k)$, with $(\mathcal{L}_j, \mathcal{L}'_j)$, $j \in [1 \mathinner{..} k]$, being the pair marking an arc in $\mathcal{M}(\mathcal{O})$ entering a state in μ_i. Thus, $\mathcal{R}$ accounts for the temporal faults relevant to the trajectories up to the initial state of the fault space x in μ_i that is exited by the arc. Moreover, $\mathcal{L}$ accounts for the temporal-fault segments of the trajectory segments from the initial state of x to the state in x that is exited by the arc (marked by $\mathcal{L}$). Hence, $\mathcal{L}'$ accounts for the temporal faults of the trajectories up to the initial state of the state in μ_{i+1} that is entered by the arc.

Let $\mathcal{E}(\mathcal{O}) = [\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_n]$, and $\mathcal{E}(\mathcal{M}(\mathcal{O})) = [\mathcal{L}(\mu_0), \mathcal{L}(\mu_1), \ldots, \mathcal{L}(\mu_n)]$. To prove eqn. (13), we have to show that $\mathcal{F}_i = \mathcal{L}_i$, $i \in [0 \mathinner{..} n]$. Based on Def. 7, $\mathcal{L}(\mu_i)$ is the alternative of the concatenated languages $\mathcal{L}_{\text{in}}(x)\mathcal{L}_{\text{out}}(x)$, where x is a state in μ_i. Besides, if $i > 0$, then $\mathcal{L}_{\text{in}}(x)$ is the alternative of the languages in second position within the pairs marking the arcs entering x; otherwise, if $i = 0$, then $\mathcal{L}_{\text{in}}(x) = \varepsilon$. Thus, $\mathcal{L}_{\text{in}}(x)$ accounts for the temporal faults relevant to the trajectories up to the arcs entering x. Similarly, if $i < n$, then $\mathcal{L}_{\text{out}}(x)$ is the alternative of the languages in first position within the pairs marking the arcs exiting x; otherwise, if $i = n$, then $\mathcal{L}_{\text{out}}(x)$ is the alternative of the languages marking the states within x. Hence, $\mathcal{L}_{\text{out}}(x)$ accounts for the temporal-fault segments relevant to the trajectory segments within x starting in the initial state of x. Hence, $\mathcal{L}(\mu_i)$ is the language of temporal faults relevant to the trajectories $T_{[i]}$ such that $Obs(T) = \mathcal{O}$; in other words, according to Def. 2, $\mathcal{L}(\mu_i) = \mathcal{F}_i$. □

5.1 Explanation Engine

The notions introduced so far, including temporal fault, explanation, and monitoring/explanation trace, all refer to a temporal observation $\mathcal{O}$ composed of a sequence of observations, namely $[o_1, \ldots, o_n]$. However, the DES being monitored generates one observation at a time, rather than in one shot. Hence, the explanation engine (that is, the software system required to explain the temporal observation) is expected to react to each new observation o by updating the current explanation based on o.

Albeit the temporal observation grows monotonically, by simple extension of the sequence of observations, the growing of the explanation turns out to be in general nonmonotonic. Specifically, if $[\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_i]$ is the explanation of the temporal observation $[o_1, \ldots, o_i]$, the explanation of $[o_1, \ldots, o_i, o_{i+1}]$ is not in general $[\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_i, \mathcal{F}_{i+1}]$, but rather, in the worst case, $[\mathcal{F}'_0, \mathcal{F}'_1, \ldots, \mathcal{F}'_i, \mathcal{F}_{i+1}]$, where $\mathcal{F}'_j \subseteq \mathcal{F}_j$, $j \in [0 \mathinner{..} i]$. More precisely, the languages that need to be restricted are in a (possibly empty) suffix of $[\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_i]$.

But, what is the cause of this nonmonotonicity? Intuitively, the extension of $\mathcal{O}$ with a new observation o is bound to make some trajectories, that were consistent with $\mathcal{O}$, no longer consistent with $\mathcal{O}' = \mathcal{O} \cup [o]$. Hence, according to eqn. (6) in Def. 2, only the trajectories T where $Obs(T) = \mathcal{O}'$ are considered in defining each language $\mathcal{F}_i$ within $\mathcal{E}(\mathcal{O}')$. The trajectories that cannot be extended based on o need therefore to be discarded, as well their temporal faults, thereby causing the possible restriction of the languages $\mathcal{F}_i$.

The pseudocode of the *Explanation Engine* is listed in Algorithm 1 (lines 1–23). It takes as input the explainer $\mathcal{X}^{\mathcal{E}}$, a temporal observation $\mathcal{O}$ of $\mathcal{X}$, the monitoring trace $\mathcal{M}$ of $\mathcal{O}$, the explanation trace $\mathcal{E}$ of $\mathcal{M}$, and a new observation o. It updates $\mathcal{O}$, $\mathcal{M}$, and $\mathcal{E}$, thereby providing the explanation of $\mathcal{O}$ extended with o. First, $\mathcal{O}$, $\mathcal{M}$, and $\mathcal{E}$ are extended in lines 9–12. Then, the set $\mathbb{X}$ of states in μ (the last node of $\mathcal{M}$ before the extension) that are not exited by any arc is computed (line 13). If $\mathbb{X}$ is not empty, then a backward pruning of $\mathcal{M}$ and $\mathcal{E}$ is carried out in lines 15–20. To this end, each state $x \in \mathbb{X}$ is removed from μ, since x terminates the trajectories that turn out to be inconsistent with the new observation o. Hence, the arcs entering these states are removed from $\mathcal{M}$ also (line 16), and $\mathcal{L}(\mu)$ in $\mathcal{E}$ is updated (line 17). Then, the pruning is propagated to the preceding node (lines 18–19) until a node including no state to be removed is found ($\mathbb{X} = \emptyset$, line 20). Eventually, $\mathcal{L}(\mu)$ is updated in any case (line 22), as some arcs exiting this node may have been cut.

Example 8 (explanation engine)**.** Shown in Table 1 is the output of the explanation engine for the temporal observation $\mathcal{O} = [val, sen, sen]$ of the DES $\mathcal{P}$ (cf. Example 2). For each index $i \in [0 \mathinner{..} 3]$ of $\mathcal{O}$, the table shows $\mathcal{O}_{[i]}$ (the prefix of $\mathcal{O}$ up to the i-th observation), the monitoring trace $\mathcal{M}(\mathcal{O}_{[i]})$, and the explanation trace $\mathcal{E}(\mathcal{M}(\mathcal{O}_{[i]}))$,

Algorithm 1 *Explanation Engine*

```
 1: procedure EXPLANATION ENGINE($\mathcal{X}^{\mathcal{E}}$, $\mathcal{O}$, $\mathcal{M}$, $\mathcal{E}$, $o$)
 2:   input $\mathcal{X}^{\mathcal{E}}$: the explainer of $\mathcal{X}$
 3:         $\mathcal{O}$: a temporal observation of $\mathcal{X}$
 4:         $\mathcal{M}$: the monitoring trace of $\mathcal{O}$
 5:         $\mathcal{E}$: the explanation trace of $\mathcal{M}$
 6:         $o$: a newly-received observation of $\mathcal{X}$
 7:   side effects: $\mathcal{O}$, $\mathcal{M}$, and $\mathcal{E}$ are updated based on $o$
 8: begin
 9:   Extend $\mathcal{O}$ by the new observation $o$
10:   Let $\mu$ denote the last node of $\mathcal{M}$ (before the extension)
11:   Extend $\mathcal{M}$ by a node $\mu'$ based on the new observation $o$
12:   Extend $\mathcal{E}$ by $\mathcal{L}(\mu')$ based on Def. 7
13:   $\mathbb{X} \leftarrow$ the set of states in $\mu$ that are not exited by any arc
14:   if $\mathbb{X} \neq \emptyset$ then
15:     repeat
16:       Remove from $\mu$ the states in $\mathbb{X}$ and their entering arcs
17:       Update $\mathcal{L}(\mu)$ in $\mathcal{E}$ based on Def. 7
18:       $\mu \leftarrow$ the node preceding $\mu$ in $\mathcal{M}$
19:       $\mathbb{X} \leftarrow$ the set of states in $\mu$ that are not exited by any arc
20:     until $\mathbb{X} = \emptyset$
21:   end if
22:   Update $\mathcal{L}(\mu)$ in $\mathcal{E}$ based on Def. 7
23: end procedure
```

which equals the explanation $\mathcal{E}(\mathcal{O}_{[i]})$ (cf. Proposition 2). When no observation has been perceived yet ($i = 0$), we have $\mathcal{E}(\mathcal{O}_{[i]}) = [\alpha_1?]$, which indicates that the fault α_1 may or may not have occurred. After the reception of the third observation (sen), backward pruning is applied to $\mathcal{M}(\mathcal{O})$ since no transition marked with a triple involving the observation sen exits the state 6 of the explainer $\mathcal{P}^{\mathcal{E}}$. As expected, the eventual explanation $\mathcal{E}(\mathcal{O})$ equals the explanation determined in Example 2 based on Def. 2.

6 Smart Knowledge Compilation

Although knowledge compilation is performed offline, assuming that the explainer is available in its entirety is impractical in real applications, because of the explosion of the number of states involved. Hence, we propose a viable approach called *smart knowledge compilation*, in which a *partial explainer* is built upfront and subsequently extended either offline, based on meaningful behavioral scenarios, or when being operated online. Analogously, the construction of the space $\mathcal{X}^*$ is impractical. Hence, hereafter, a notation like $\langle x, t, x' \rangle \in \mathcal{X}^*$ does not assume that $\mathcal{X}^*$ is available: it is only a shorthand for stating that the component transition t is triggerable at the state x of $\mathcal{X}$.

Definition 8 (partial explainer)**.** *Let $\mathcal{X}^{\mathcal{E}}$ be the explainer of $\mathcal{X}$. A* partial explainer *of $\mathcal{X}$, denoted $\mathcal{X}^e$, is a connected subgraph of $\mathcal{X}^{\mathcal{E}}$ that includes the initial state of $\mathcal{X}^{\mathcal{E}}$.*

Example 9 (partial explainer)**.** With reference to the explainer $\mathcal{P}^{\mathcal{E}}$ displayed in Fig. 4, a partial explainer $\mathcal{P}^e$ is shown in Fig. 6. In particular, $\mathcal{P}^e$ is a *prefix* of $\mathcal{P}^{\mathcal{E}}$ at distance 1, that is, the subgraph of $\mathcal{P}^{\mathcal{E}}$ that includes all the states and transitions that can be reached (from the initial state) by one (dashed) transition.

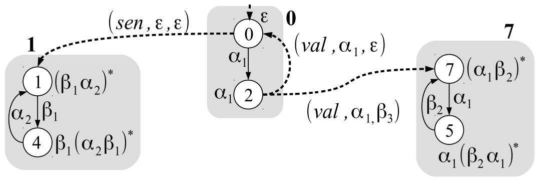


Figure 6: Partial explainer $\mathcal{P}^e$, prefix of $\mathcal{P}^{\mathcal{E}}$ (distance = 1).

Once an initial partial explainer $\mathcal{X}^e$ is somehow generated, for instance a prefix of the (whole) explainer $\mathcal{X}^{\mathcal{E}}$, it can be *upgraded* in several ways. One such way is by exploiting *behavioral scenarios*, which are particular evolutions of the DES that are required to be explained efficiently.

Definition 9 (behavioral scenario)**.** *Let $\mathcal{X}$ be a DES and $\mathbb{T}$ a subset of the component transitions in $\mathcal{X}$. A* behavioral scenario *of $\mathcal{X}$ is a pair $\mathcal{S} = (\mathbb{T}, \mathcal{L})$, where $\mathcal{L}$ is a regular language on $\mathbb{T}$.*

Example 10 (behavioral scenario)**.** A scenario where the only malfunction is the valve being stuck closed can be defined as $\mathcal{S} = (\mathbb{T}, \mathcal{L})$, where $\mathbb{T} = \{s_3, s_4, v_1, v_2, v_3, v_4, v_7, v_8\}$ and $\mathcal{L} = v_3 v_3^+$ (repetition of v_3 at least twice).

To upgrade a partial explainer $\mathcal{X}^e$ so that it embeds the set of temporal observations generated by a scenario $\mathcal{S}$ (the observation language of $\mathcal{S}$, namely $OBS(\mathcal{S})$), we need to synchronize $\mathcal{S}$ with the behavior of $\mathcal{X}$, as described below.

Definition 10 (abduction)**.** *Let $\mathcal{S} = (\mathbb{T}, \mathcal{L})$ be a scenario of $\mathcal{X}$. The* restriction *of a trajectory T in $\mathcal{X}^*$ on $\mathbb{T}$ is the sequence $T_{\mathbb{T}} = [t \mid t \in T, t \in \mathbb{T}]$. The* abduction *of $\mathcal{S}$, $\mathcal{X}^*_{\mathcal{S}}$, is a DFA whose language is the set $\{T \mid T \in \mathcal{X}^*, T_{\mathbb{T}} \in \mathcal{L}\}$.*

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

Example 11 (abduction)**.** With reference to the behavioral scenario $\mathcal{S}$ defined in Example 10, shown in Fig. 7 are the DFA recognizing $\mathcal{S}$ (left) and the abduction $\mathcal{P}^*_{\mathcal{S}}$ (right). Each state of $\mathcal{P}^*_{\mathcal{S}}$ is a pair (p, d), where p is a state of $\mathcal{P}^*$ and d is a state of the DFA. A state is final when d is final ($d = 2$).

The next step is to distill the observation language of the abduction, namely the set of temporal observations generated by the trajectories in the abduction (the *observation pattern*).

Definition 11 (observation pattern)**.** *Let $\mathcal{X}$ be a DES and* **O** *the domain of observations involved in the mapping table $\mu(\mathcal{X})$. An* observation pattern *$\mathcal{O}^*$ of $\mathcal{X}$ is a DFA whose language is a set of strings on* **O**.

Def. 11 is general in nature. Still, meaningful observation patterns can be drawn from abductions. Specifically, each symbol t marking a transition $\langle a, t, a' \rangle$ in an abduction $\mathcal{X}^*_{\mathcal{S}}$ is replaced with a (possibly ε) observation o, where

Figure 7: DFA of scenario $\mathcal{S}$ (left) and abduction $P^*_{\mathcal{S}}$ (right).

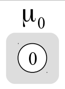
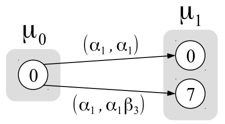
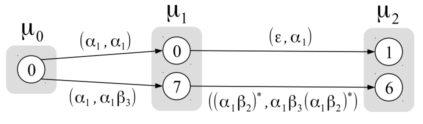
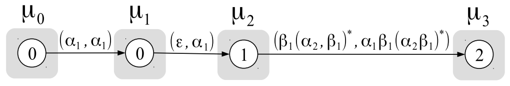
i	$\mathcal{O}_{[i]}$	*Monitoring trace* $\mathcal{M}\left(\mathcal{O}_{[i]}\right)$	*Explanation trace* $\mathcal{E}\left(\mathcal{M}\left(\mathcal{O}_{[i]}\right)\right)$
0	$[\,]$		$[\alpha_1?]$
1	$[val]$		$[\alpha_1, (\alpha_1\alpha_1?\|\alpha_1\beta_3((\alpha_1\beta_2)^*\|\alpha_1(\beta_2\alpha_1)^*))]$
2	$[val, sen]$		$[\alpha_1, (\alpha_1\|\alpha_1\beta_3(\alpha_1\beta_2)^*), (\alpha_1((\beta_1\alpha_2)^*\|\beta_1(\alpha_2\beta_1)^*)\|\alpha_1\beta_3(\alpha_1\beta_2)^*)]$
3	$[val, sen, sen]$		$[\alpha_1, \alpha_1, \alpha_1\beta_1(\alpha_2\beta_1)^*, \alpha_1\beta_1(\alpha_2\beta_1)^*]$

Table 1: Temporal output of the explanation engine for the temporal observation $\mathcal{O} = [val, sen, sen]$ of the DES $\mathcal{P}$.

$(t, o, f) \in \mu(\mathcal{X})$. The resulting NFA is then determinized into an equivalent (possibly minimized) DFA, which is by definition the observation pattern of the scenario $\mathcal{S}$, namely $\mathcal{O}^*_{\mathcal{S}}$. Remarkably, the language of $\mathcal{O}^*_{\mathcal{S}}$ is the set of temporal observations associated with the set of trajectories in the abduction, with each trajectory being a way the scenario $\mathcal{S}$ manifests itself in $\mathcal{X}^*$.

Example 12 (observation pattern)**.** With reference to the scenario $\mathcal{S}$ defined in Example 10 and the abduction $\mathcal{P}^*_{\mathcal{S}}$ in Fig. 7, shown in Fig. 8 is the observation pattern $\mathcal{O}^*_{\mathcal{S}}$.

6.1 Explainer Upgrade

To upgrade a partial explainer $\mathcal{X}^e$ based on an observation pattern $\mathcal{O}^*$, Algorithm 2 is used (lines 1–30). Each state x^e in $\mathcal{X}^e$ is assumed to be marked with a *labeling set* (initially empty), denoted $\Lambda(x^e)$, which contains states of $\mathcal{O}^*$. This serves to synchronize $\mathcal{X}^e$ with $\mathcal{O}^*$ avoiding duplications of $\mathcal{X}^e$ states, as well as endless loops caused by cycles in $\mathcal{O}^*$. If a state ω in $\Lambda(x^e)$ is unmarked, it means that the transitions exiting ω in $\mathcal{O}^*$ need to be synchronized with the transitions exiting x^e in $\mathcal{X}^e$. In case a transition is missing in $\mathcal{X}^e$, it is created, possibly along with its target state, a fault space, which is marked with the relevant labeling set (line 20). Once all transitions exiting ω in $\mathcal{O}^*$ have been processed, ω is marked in $\Lambda(x^e)$ (line 26). The processing terminates when there is no unmarked ω in any labeling set.

Example 13 (explainer upgrade)**.** Let $\mathcal{P}^e$ be the partial explainer defined in Example 9 and shown in Fig. 6. Let $\mathcal{O}^*_{\mathcal{S}}$ be the observation pattern displayed in Fig. 8. The partial explainer resulting from the application of Algorithm 2 on $\mathcal{P}^e$ and $\mathcal{O}^*_{\mathcal{S}}$ is shown in Fig. 9.

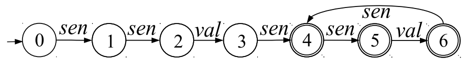


Figure 8: Observation pattern $\mathcal{O}^*_{\mathcal{S}}$ (cf. $\mathcal{P}^*_{\mathcal{S}}$ in Fig. 7).

To what extent does the language of a partial explainer change after being upgraded based on an observation pattern? A formal answer is provided by Proposition 3.

Proposition 3. *Let $\mathcal{X}^e$ be a partial explainer, $\mathcal{O}^*$ an observation pattern for $\mathcal{X}$, and $\mathcal{X}'^e$ the partial explainer resulting from the application of the procedure* Explainer Upgrade *to $\mathcal{X}^e$ and $\mathcal{O}^*$ (Algorithm 2). The following relationship among observation languages holds:*

$$OBS\left(\mathcal{X}'^e\right) \supseteq OBS\left(\mathcal{X}^e\right) \cup \left(OBS\left(\mathcal{O}^*\right) \cap OBS\left(\mathcal{X}^*\right)\right). \quad (15)$$

Intuitively, relationship (15) is grounded on two facts. First, in general, not all the temporal observations in $OBS(\mathcal{O}^*)$ are consistent with the space of $\mathcal{X}$ and, hence, the observation language of $\mathcal{X}'^e$ will include the additional temporal observations in $OBS(\mathcal{O}^*) \cap OBS(\mathcal{X}^*)$. Second, $OBS(\mathcal{X}'^e)$ is bound to include (possibly an infinite number of) new temporal observations not in this intersection. For example, if $\mathcal{O} = [a]$ and $\mathcal{X}^{\mathcal{E}}$ involves an auto-transition on the initial state marked by $(a, \mathcal{L}, f)$, then $OBS(\mathcal{X}'^e)$ will include not only $[a]$ but all the strings in the infinite set $[a, aa, aaa, \ldots]$.

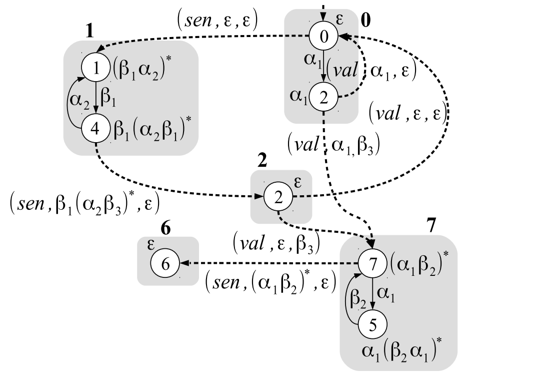


Figure 9: Partial explainer obtained by upgrading $\mathcal{P}^e$ in Fig. 6 based on the observation pattern $\mathcal{O}^*_{\mathcal{S}}$ in Fig. 8.

Algorithm 2 *Explainer Upgrade*

```
1: procedure EXPLAINER UPGRADE($\mathcal{X}^e$, $\mathcal{O}^*$)
2:   input $\mathcal{X}^e$: a partial explainer of $\mathcal{X}$, with initial state $x_0^e$
3:         $\mathcal{O}^*$: an observation pattern for $\mathcal{X}$, with initial state $\omega_o$
4:   side effects: $\mathcal{X}^e$ is upgraded based on $\mathcal{O}^*$
5: begin
6:   Insert $\omega_0$ into the (initially empty) labeling set $\Lambda(x_0^e)$
7:   repeat
8:     Let $\Lambda(x_e)$ be a set including an unmarked pattern state $\omega$
9:     for all unmarked pattern state $\omega \in \Lambda(x_e)$ do
10:      for all transition $\langle \omega, o, \omega' \rangle$ in $\mathcal{O}^*$ do
11:        if $\exists$ a transition exiting $x^e$ marked with $(o, \mathcal{L}, f)$ then
12:          for all transition $\langle x^e, (o, \mathcal{L}, f), x'^e \rangle$ in $\mathcal{X}^e$ do
13:            Insert $\omega'$ into $\Lambda(x'^e)$, unless $\omega' \in \Lambda(x'^e)$ already
14:          end for
15:        else
16:          for all $x \in x^e, \langle x, t, x' \rangle \in \mathcal{X}^*, (t, o, f) \in \mu(\mathcal{X})$ do
17:            Let $x'^e$ denote the fault space of $x'$, namely $\mathcal{X}^*_{x'}$
18:            if $\mathcal{X}^e$ does not include the state $x'^e$ then
19:              Create the state $x'^e = \mathcal{X}^*_{x'}$ in $\mathcal{X}^e$
20:              Mark $x'^e$ with the labeling set $\{\omega'\}$ (singleton)
21:            end if
22:            Create the transition $\langle x_e, (o, \mathcal{L}(x), f), x'^e \rangle$ in $\mathcal{X}^e$
23:          end for
24:        end if
25:      end for
26:      Mark the pattern state $\omega$ within the labeling set $\Lambda(x^e)$
27:    end for
28:  until there is no labeling set $\Lambda$ including an unmarked state
29:  Empty all the nonempty labeling sets $\Lambda$ in $\mathcal{X}^e$
30: end procedure
```

Corollary 3.1. *With reference to Proposition 3, if $\mathcal{O}^*$ is the observation pattern of a behavioral scenario of $\mathcal{X}$, then*

$$OBS\left(\mathcal{X}'^e\right) \supseteq OBS\left(\mathcal{X}^e\right) \cup OBS\left(\mathcal{O}^*\right). \qquad (16)$$

Since an abduction is a subspace of $\mathcal{X}^*$ (including the initial state), if the observation pattern $\mathcal{O}^*$ is distilled from the abduction $\mathcal{X}^*_{\mathcal{S}}$ of a scenario $\mathcal{S}$, then $\mathcal{O}^*$ is sound, that is, $OBS(\mathcal{O}^*) \subseteq OBS(\mathcal{X}^*)$. This is why in eqn. (16) there is no need for the intersection $OBS(\mathcal{O}^*) \cap OBS(\mathcal{X}^*)$.

6.2 Smart Explanation Engine

The *Explanation Engine* specified in Algorithm 1 needs to be revised when the explainer is partial. Specifically, before line 11 of Algorithm 1, we need to be sure that the transition function of each state in μ is complete as far as the observation o is concerned.

The new algorithm, called *Smart Explanation Engine* (Algorithm 3) differs from Algorithm 1 in two ways. First, it takes as input a *partial* explainer $\mathcal{X}^e$. Second, and more to the point, it includes a new fragment of pseudocode (lines 5–15) aimed at upgrading $\mathcal{X}^e$ based on o, specifically for the possibly missing transitions involving the observation o and exiting the states in μ.

To speed up the processing, each state in $\mathcal{X}^e$ may be marked with the set $\mathbb{O}$ of observations for which the transition function is complete. This way, if $o \in \mathbb{O}$, lines 6–14 may be skipped.

Algorithm 3 *Smart Explanation Engine*

```
1: procedure SMART EXPLANATION ENGINE($\mathcal{X}^e, \mathcal{O}, \mathcal{M}, \mathcal{E}, o$)
...         # Similar declarations as in lines 2–7 of Algorithm 1
2: begin
3:   Extend $\mathcal{O}$ by the new observation $o$
4:   Let $\mu$ denote the last node of $\mathcal{M}$ (before the extension)
5:   for all $x^e \in \mu$ do
6:     if there is no transition $\langle x^e, o, x'^e \rangle$ in $\mathcal{X}^e$ then
7:       for all $x \in x^e, \langle x, t, x' \rangle \in \mathcal{X}^*, (t, o, f) \in \mu(\mathcal{X})$ do
8:         Let $x'^e$ denote the fault space of $x'$, namely $\mathcal{X}^*_{x'}$
9:         if $x'^e \notin \mathcal{X}^e$ then
10:          Create the state $x'^e = \mathcal{X}^*_{x'}$ in $\mathcal{X}^e$
11:        end if
12:        Insert the transition $\langle x_e, (o, \mathcal{L}(x), f), x'^e \rangle$ into $\tau^e$
13:      end for
14:    end if
15:  end for
...              # Same code as in lines 11–22 of Algorithm 1
16: end procedure
```

Example 14 (smart explanation engine)**.** Let $\mathcal{P}^e$ be the partial explainer displayed in Fig. 9 and $\mathcal{O} = [val, sen, sen]$ the temporal observation defined in Example 8, whose explanation is traced in Table 1. The iterated application of *Smart Application Engine* on $\mathcal{O}$ based on $\mathcal{P}^e$ does not change $\mathcal{P}^e$. By contrast, if $\mathcal{O} = [sen, val, sen]$, then *Smart Application Engine* will upgrade $\mathcal{P}^e$ as shown in Fig. 10.

7 Conclusion

This paper deals with monitoring-based diagnosis of DESs by proposing a shift from a set-oriented (hence, temporally-free) to a temporal-oriented perspective. A temporal-oriented diagnosis output, represented as a (finite) regular expression over the alphabet of faults, supports explainability, which is further enhanced by performing a backward pruning every time a new observation is perceived. This technique guarantees that all the candidates relevant to any prefix of the temporal observation received so far are consistent with the whole temporal observation received so far.

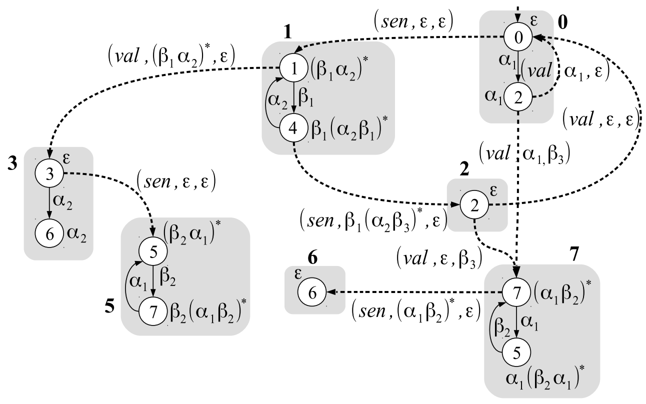


Figure 10: Partial explainer obtained by the iterated application of Algorithm 3, starting from the partial explainer in Fig. 9 and based on the temporal observation $[sen, val, sen]$.

Backward pruning and smart knowledge compilation take inspiration from (Bertoglio, Lamperti, and Zanella 2019a; Bertoglio, Lamperti, and Zanella 2019b; Bertoglio et al. 2019); however, the quoted works adopt a set-oriented perspective.

The diagnoser approach (Sampath et al. 1995), besides being set-oriented, does not cope with a possibly infinite number of possibly infinite trajectories that entail the given temporal observation. In fact, the diagnoser approach assumes that both the language of the transitions and the language of the observable events of the DES are live and that the faulty transitions are unobservable, whereas such assumptions are relaxed here. Consequently, while according to the diagnoser approach there does not exist any unobservable behavioral cycle, and hence, there does not exist any cycle of faults, both such cycles are allowed in the current approach. Therefore, in this paper the regular expressions relevant to the occurrence of faults can represent an unbounded number of iterations, while this is not needed in case the temporal fault characterization were adopted by the diagnoser approach (or by approaches making the same assumptions).

Our temporal perspective in diagnosis of DESs is relevant to the diagnosis output, while the component models are untimed. Some contributions in the literature, instead, added temporal information to the DES models, such as (Hashtrudi Zad, Kwong, and Wonham 2005), where the clock tick is an extra input signal: thus, the DES model has changed while a diagnosis is still a set of faults.

A DES model that, instead of being an automaton, is a causal network is adopted in (Darwiche and Provan 1996), which deals with diagnosis of dynamic systems governed by a discrete controller that issues commands at discrete time points. In order to incorporate temporal information in the model, a causal network is created for each time point in the time-step cycle of the system, and each network is connected with the next time-point network. Once again, the notion of diagnosis has not changed.

Other contributions in the literature are meant to generalize the notion of a fault in DESs. This is the case with (Jiang and Kumar 2004), where a fault is defined as the violation of an LTL-specification (among several given ones). However, a diagnosis is still a set of faults, while in this paper a diagnosis is a set of temporal faults.

In (Jéron et al. 2006), the notion of a fault is generalized to a pattern, this being a DFA that can represent the ordered occurrences of multiple faults, the multiple occurrences of the same fault, etc. It is tempting to speculate that a diagnosis output consisting in a set of temporal faults resembles diagnosis with supervision patterns as a pattern enables the detection of a specific language of transitions and, therefore, the detection of a specific language of faulty transitions also. In other words, the supervision pattern approach can find out whether there exists a trajectory that both implies the given temporal observation and complies with the given (pattern) language. Notice that there may exist several other trajectories that imply the temporal observation while producing sequences of faults that do not belong to the given (pattern) language: the supervision pattern approach does not produce any output about them. By contrast, the approach described in this paper is not given any automaton upfront recognizing a language, instead it produces a regular expression representing the language of the faults of all the trajectories that imply the given sequence of observations. Moreover, the output of the supervision pattern approach clarifies whether the pattern has occurred; however, it does not compute the number of its occurrences, nor does it show the reciprocal order of these occurrences and those of individual faults within the trajectories implying the temporal observation. In the view of the current paper, instead, if a fault is associated with a pattern, this can be part of a temporal fault as all other faults are. In summary, temporal faults are orthogonal to the classification of faults, be they either 'simple' or somehow 'complex' as in (Jéron et al. 2006; Lamperti and Zanella 2011; Lamperti and Zhao 2014).

A challenging idea for future research is the generalization of the notion of explainability of DESs proposed in this paper. Specifically, the number of observations after which the sets of temporal faults are required to be finite for a DES to be enumerably explainable could be larger than one, with various degrees of unexplainability being envisaged. Moreover, explainability could be compared with other properties of DESs, such as *diagnosability* (Sampath et al. 1995; Pencolé 2004; Schumann and Huang 2008) and *manifestability* (Dague, He, and Ye 2019). Finally, finding formal conditions for the explainability of a DES and extending explanatory diagnosis to complex DESs (Lamperti and Quarenghi 2016; Lamperti and Zhao 2016a; Lamperti and Zhao 2016b; Lamperti, Zanella, and Zhao 2018b) are further interesting topics for future work.

Acknowledgments

This work was supported in part by Regione Lombardia (project Smart4CPPS, Linea Accordi per Ricerca, Sviluppo e Innovazione, POR-FESR 2014-2020 Asse I) and by the National Natural Science Foundation of China (grant number 61972360).

References

Basile, F. 2014. Overview of fault diagnosis methods based on Petri net models. In *Proceedings of the 2014 European Control Conference, ECC 2014*, 2636–2642.

Bertoglio, N.; Lamperti, G.; Zanella, M.; and Zhao, X. 2019. Twin-engined diagnosis of discrete-event systems. *Engineering Reports* 1:1–20.

Bertoglio, N.; Lamperti, G.; and Zanella, M. 2019a. Intelligent diagnosis of discrete-event systems with preprocessing of critical scenarios. In Czarnowski, I.; Howlett, R.; and Jain, L., eds., *Intelligent Decision Technologies 2019*, volume 142 of *Smart Innovation, Systems and Technologies*. Springer, Singapore. 109–121.

Bertoglio, N.; Lamperti, G.; and Zanella, M. 2019b. Temporal diagnosis of discrete-event systems with dual knowledge compilation. In Holzinger, A.; Kieseberg, P.; Weippl, E.; and Tjoa, A. M., eds., *Machine Learning and Knowledge Extraction*, volume 11713 of *Lecture Notes in Computer Science*. Springer, Berlin. 333–352.

Brand, D., and Zafiropulo, P. 1983. On communicating finite-state machines. *Journal of the ACM* 30(2):323–342.

Brzozowski, J., and McCluskey, E. 1963. Signal flow graph techniques for sequential circuit state diagrams. *IEEE Transactions on Electronic Computers* EC-12(2):67–76.

Cassandras, C., and Lafortune, S. 2008. *Introduction to Discrete Event Systems*. New York: Springer, second edition.

Cong, X.; Fanti, M.; Mangini, A.; and Li, Z. 2018. Decentralized diagnosis by Petri nets and integer linear programming. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48(10):1689–1700.

Dague, P.; He, L.; and Ye, L. 2019. How to be sure a faulty system does not always appear healthy? *Innovations in Systems and Software Engineering*.

Darwiche, A., and Provan, G. 1996. Exploiting system structure in model-based diagnosis of discrete-event systems. In *7th International Workshop on Principles of Diagnosis (DX 1996)*, 95–105.

Hamscher, W.; Console, L.; and de Kleer, J., eds. 1992. *Readings in Model-Based Diagnosis*. San Mateo, CA: Morgan Kaufmann.

Hashtrudi Zad, S.; Kwong, R.; and Wonham, W. 2005. Fault diagnosis in discrete-event systems: incorporating timing information. *IEEE Transactions on Automatic Control* 50(7):1010–1015.

Hopcroft, J.; Motwani, R.; and Ullman, J. 2006. *Introduction to Automata Theory, Languages, and Computation*. Reading, MA: Addison-Wesley, third edition.

Jéron, T.; Marchand, H.; Pinchinat, S.; and Cordier, M. 2006. Supervision patterns in discrete event systems diagnosis. In *Workshop on Discrete Event Systems (WODES 2006)*, 262–268. Ann Arbor, MI: IEEE Computer Society.

Jiang, S., and Kumar, R. 2004. Failure diagnosis of discrete-event systems with linear-time temporal logic specifications. *IEEE Transactions on Automatic Control* 49(6):934–945.

Lamperti, G., and Quarenghi, G. 2016. Intelligent monitoring of complex discrete-event systems. In Czarnowski, I.; Caballero, A.; Howlett, R.; and Jain, L., eds., *Intelligent Decision Technologies 2016*, volume 56 of *Smart Innovation, Systems and Technologies*. Springer International Publishing Switzerland. 215–229.

Lamperti, G., and Zanella, M. 2011. Context-sensitive diagnosis of discrete-event systems. In Walsh, T., ed., *Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, volume 2, 969–975. Barcelona, Spain: AAAI Press.

Lamperti, G., and Zhao, X. 2014. Diagnosis of active systems by semantic patterns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44(8):1028–1043.

Lamperti, G., and Zhao, X. 2016a. Diagnosis of complex active systems with uncertain temporal observations. In Buccafurri, F.; Holzinger, A.; Tjoa, A. M.; and Weippl, E., eds., *Availability, Reliability, and Security in Information Systems*, volume 9817 of *Lecture Notes in Computer Science*. Springer International Publishing AG Switzerland. 45–62.

Lamperti, G., and Zhao, X. 2016b. Viable diagnosis of complex active systems. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016)*, 457–462.

Lamperti, G.; Zanella, M.; and Zhao, X. 2018a. *Introduction to Diagnosis of Active Systems*. Springer, Cham.

Lamperti, G.; Zanella, M.; and Zhao, X. 2018b. Knowledge compilation techniques for model-based diagnosis of complex active systems. In Holzinger, A.; Kieseberg, P.; Tjoa, A. M.; and Weippl, E., eds., *Machine Learning and Knowledge Extraction*, volume 11015 of *Lecture Notes in Computer Science*. Springer, Cham. 43–64.

McIlraith, S. 1998. Explanatory diagnosis: conjecturing actions to explain observations. In *Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR 1998)*, 167–177. Trento, I: Morgan Kaufmann, S. Francisco, CA.

Pencolé, Y.; Steinbauer, G.; Mühlbacher, C.; and Travé-Massuyès, L. 2017. Diagnosing discrete event systems using nominal models only. In *28th International Workshop on Principles of Diagnosis (DX 2017)*, 169–183.

Pencolé, Y. 2004. Diagnosability analysis of distributed discrete event systems. In *Sixteenth European Conference on Artificial Intelligence (ECAI 2004)*, 43–47.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artificial Intelligence* 32(1):57–95.

Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K.; and Teneketzis, D. 1995. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control* 40(9):1555–1575.

Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K.; and Teneketzis, D. 1996. Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology* 4(2):105–124.

Schumann, A., and Huang, J. 2008. A scalable jointree algorithm for diagnosability. In *Twenty-Third National Conference on Artificial Intelligence (AAAI 2008)*, 535–540.

Struss, P. 1997. Fundamentals of model-based diagnosis of dynamic systems. In *Fifteenth International Joint Conference on Artificial Intelligence (IJCAI 1997)*, 480–485.