

Machine Learning for Stream-Based Diagnosis

Raik Hipler[✉] and Martin Leucker[✉]

University of Luebeck, Luebeck, Germany
{hipler,leucker}@isp.uni-luebeck.de

Abstract

A major challenge in model-based diagnosis is the modeling of systems. Previous research has explored how machine learning can be applied to learn normative system behavior. We propose to apply this technique to a recent approach that integrates model-based diagnosis and runtime verification within the stream specification language LOLA.

Introduction and Motivation

Model-based diagnosis (MBD) [10] is a well-known technique to localize faults within a system that is not behaving as intended. A *system* consists of a set of *components* COMPS and a set of formulas called the *system description* SD . The latter primarily consists of formulas $\neg \text{AB}(c) \rightarrow \text{beh}_c$ which encode “if component $c \in \text{COMPS}$ is not abnormal, its normal (intended) behavior beh_c must hold”, with the subformula beh_c often defining the output of c in terms of c ’s inputs. Given an observation OBS (e.g., input and output values of observed components), the goal of MBD is to localize the fault (if there is one) by identifying the components causing the misbehavior. A set of components $\Delta \subseteq \text{COMPS}$ is called a *diagnosis* if the observation can be explained by assuming components $c \in \Delta$ to be abnormal and all other components to be normal, i.e., $SD \cup \text{OBS}$ is consistent under that assumption. See [6, Ex. 1–2] for a full example.

A recent approach [6] integrates MBD with *runtime verification (RV)* [7] through the stream specification language *LOLA* [1]. In *LOLA*, for each time instant, the values of streams are defined in terms of the values of (other) streams at possibly different time instants; inputs are received on input streams. The framework in [6] is based on [4] to support *uncertainties*, that is, nondeterministic inputs (e.g., noise), by encoding stream values symbolically as constraints. Such constraints allow reasoning over the stream values (e.g., via an SMT solver). Additionally, the specification can be enriched by a special *assumption* stream, which holds by definition [4]. This allows one to encode background knowledge, enabling more precise reasoning (see [6, Ex. 3–4]). In [6], system descriptions are encoded as assumptions and observations are received on input streams (see [6, Ex. 5]). The former employ dedicated streams indicating that a specific component is abnormal at a specific time instant. In the presence of a fault, in order to uphold the consistency of system description and observations, certain combinations of such component streams must therefore indicate the corresponding components to be abnormal at certain time instants, which can be inferred through SMT-backed reasoning. The framework allows one to specify monitors for fault detection [4, 5] as well as diagnosis problems in a unified encoding.

A major challenge of MBD is to model the system, i.e., precisely and explicitly defining each component’s normal behavior [8]. Since a system description consists of logical formulas over discrete observations, this can be especially difficult for sensors in cyber-physical systems where data is typically continuous and it is not obvious how to encode the normal behavior. E.g., should a pressure reading of 45 psi be considered high? Where should the threshold lie? How should readings of multiple sensors be interpreted jointly? A recent approach [9] utilizes machine learning (ML) to extract symbolic representations in order to derive the normal behavior of components. The derived formulas are ensured to conform to the commonly assumed *weak-fault model (WFM)* [11] which implies that subset-minimal diagnoses are representative of all diagnoses. We propose to apply this approach in the aforementioned *LOLA* diagnosis framework.

Proposed Integration and Future Research

The core idea of [9] is as follows. A data-driven model is trained (in this case a CatVAE), such that, for a given observation, it estimates (i) the *state* the system is likely to be currently operating in (e.g., *heating* or *cooling*), and (ii) a *residual*, that is, an indicator of inconsistency (the deviation from the expected behavior). Both are discretized; the state is chosen from a finite domain, and the residual is abstracted to a Boolean value r (*true* if the deviation is small). Because the observations correspond to certain components $C \subseteq \text{COMPS}$, if r is *false* for a certain state, at least one $c \in C$ must be abnormal. This makes it possible to derive WFM-conform formulas $h_C \rightarrow (s_i \rightarrow r)$ with $h_C := \bigwedge_{c \in C} \neg \text{AB}(c)$, encoding “if all $c \in C$ are normal, then the system being in state s_i (according to the model) implies (very close to) expected behavior.” The set of such formulas constitutes a system description to be used in diagnosis. The model is trained unsupervised on normal behavior (possibly obtained from a simulation).

To support this technique in the LOLA framework [6], the model has to be integrated at the stream level. Instead of receiving observations on input streams, they are given to the model, which in turn delivers the current state and residual on input streams. The formulas from above can then be used as system description in the LOLA assumption, enabling the application of the diagnosis notions from [6]. While ML-derived system descriptions inevitably sacrifice the formal soundness guarantee of the LOLA framework, they enable diagnosis in domains where manual modeling is infeasible, and the formal inference properties of LOLA remain sound with respect to the system description. However, there are many open questions for future research:

In [9], a diagnosis only corresponds to a single observation. While support for multiple observations has been considered [12], the observations are still independent of each other and the diagnoses do not reason about time, i.e., they only identify *which* components explain all observations but not *when* those components would have been abnormal. Since the LOLA framework naturally supports temporal diagnosis [2] as well as stateful systems whose behavior may also depend on past inputs, the model should be able to estimate state and residual for observations from multiple subsequent time instants, e.g., the k most recent instants. Alternatively (or additionally), alongside the $h_C \rightarrow (s_i \rightarrow r)$ formulas, the system description could encompass rules that reason about state transitions, e.g., that after state s_i the system must either stay in s_i or switch to s_j but not to s_k ; otherwise one component from C must be abnormal. However, since the states obtained from the model in [9] are unlabeled, they do not have intuitive meaning, and must be manually labeled (symbol grounding [3]), e.g., by finding common patterns.

One aspect unexplored in [9] is that a single model can only construct rules for a fixed $C \subseteq \text{COMPS}$ because observations given to the model always correspond to the same components. This limits diagnostic precision since the components in C cannot be distinguished. By using multiple C that only partially overlap, the actual diagnosis can be more precisely pinpointed. However, this requires possibly more observable values within the system, and the training of multiple (albeit smaller) models, each of which is effectively responsible for one subsystem.

Furthermore, the previously mentioned support for nondeterministic inputs in the LOLA framework could be extended towards the model’s state estimation, such that more than one state is considered possible given a particular observation.

A different direction could be to dynamically train the network during runtime. If a monitor (also specified in LOLA) is not triggered, correct behavior is assumed and the current observation is used to further train the model. When a new system state is discovered, new rules could be added to the assumption. Conversely, when a monitor is triggered, the observation implies a fault, which could enable the application of learning techniques besides unsupervised learning.

To conclude, we believe that integrating machine learning with the LOLA diagnosis framework is a worthwhile endeavor for improving its practical applicability.

References




- [1] D'Angelo, B., Sankaranarayanan, S., Sánchez, C., Robinson, W., Finkbeiner, B., Sipma, H.B., Mehrotra, S., Manna, Z.: LOLA: runtime monitoring of synchronous systems. In: Proc. of the 12th Int'l Symp. of Temporal Representation and Reasoning (TIME'05). pp. 166–174. IEEE Computer Society (2005). <https://doi.org/10.1109/TIME.2005.26>
- [2] Friedrich, G., Lackinger, F.: Diagnosing Temporal Misbehavior. In: IJCAI. pp. 1116–1122 (1991)
- [3] Harnad, S.: The symbol grounding problem. *Physica D: Nonlinear Phenomena* **42**(1), 335–346 (1990). [https://doi.org/10.1016/0167-2789\(90\)90087-6](https://doi.org/10.1016/0167-2789(90)90087-6)
- [4] Hipler, R., Kallwies, H., Leucker, M., Montali, M., Sánchez, C., Winkler, S.: Symbolic runtime verification for monitoring under uncertainties and assumptions. *Information and Software Technology* **191**, 108004 (2026). <https://doi.org/10.1016/j.infsof.2025.108004>
- [5] Hipler, R., Kallwies, H., Leucker, M., Sánchez, C.: General anticipatory runtime verification. In: Proc. 36th CAV. LNCS, vol. 14682, pp. 133–155 (2024). https://doi.org/10.1007/978-3-031-65630-9_7
- [6] Hipler, R., Leucker, M., Rodler, P.: A unified framework for runtime verification and model-based diagnosis in LOLA (2026), unpublished
- [7] Leucker, M., Schallhart, C.: A brief account of runtime verification. *J. Logic Algebr. Progr.* **78**(5), 293–303 (2009)
- [8] Lucas, P., Dal, G., Hommersom, A., Grievink, G.: Model-based probabilistic diagnosis in large cyberphysical systems. In: Proceedings of the European Conference of the PHM Society 2024. vol. 8, pp. 643–654 (2024). <https://doi.org/10.36001/phme.2024.v8i1.4033>, 8th European Conference of the Prognostics and Health Management Society 2024
- [9] Moddemann, L., Steude, H.S., Diedrich, A., Niggemann, O.: Discret2di - deep learning based discretization for model-based diagnosis. *IFAC-PapersOnLine* **58**(4), 640–645 (2024). <https://doi.org/https://doi.org/10.1016/j.ifacol.2024.07.291>, 12th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2024
- [10] Reiter, R.: A Theory of Diagnosis from First Principles. *Artificial Intelligence* **32**(1), 57–95 (1987)
- [11] Rodler, P.: How Should I Compute My Candidates? A Taxonomy and Classification of Diagnosis Computation Algorithms. In: ECAI 2023, pp. 1986–1993. IOS Press (2023)
- [12] Tai, R., Ouyang, D., Zhang, L.: Two algorithms for improving model-based diagnosis using multiple observations and deep learning. *Neural Networks* **185**, 107185 (2025). <https://doi.org/10.1016/j.neunet.2025.107185>

Appendix

Starting from the next page, the remainder of this document consists of the unpublished paper [6].

A Unified Framework for Runtime Verification and Model-Based Diagnosis in LOLA

—UNPUBLISHED—

Raik Hipler¹, Martin Leucker¹, and Patrick Rodler²

¹ University of Luebeck, Luebeck, Germany
{hipler,leucker}@isp.uni-luebeck.de

² University of Klagenfurt, Klagenfurt, Austria
patrick.rodler@aau.at

Abstract. Runtime verification has proven effective for detecting failures, while model-based diagnosis is a powerful method for identifying their underlying causes. Combining these two techniques is therefore highly valuable, as it can ultimately support mitigation measures for detected failures. This paper proposes an integrated approach that specifies both runtime verification and diagnostic information within the well-known stream-based language LOLA. The resulting unified framework offers the expression of various verification and diagnosis scenarios, as well as the combination of symbolic, stream-based reasoning under uncertainty with diagnostic inference for automated fault localization.

Keywords: Runtime Verification · LOLA · Uncertainties · Assumptions · Model-Based Diagnosis · Multiple Observations · Temporal Diagnosis

1 Introduction

Beyond the correctness of safety-critical systems, their resilience is of paramount importance. In particular, systems must react appropriately to faults occurring during operation. To this end, failures must first be *detected*, i.e., the system must recognize that something is going wrong, and then *localized*, meaning that the root cause of the malfunction (e.g., a faulty component) must be determined.

Runtime verification (RV) [21] is actively researched and well suited for detecting deviations from the specified behavior of a system during runtime. Similarly, *model-based diagnosis (MBD)* [26] is a well-founded, principled framework for localizing faults when a system no longer behaves as intended. Integrating RV with MBD promises a synergistic combination: automated fault detection and systematic fault localization in a live system monitoring framework.

Early work by [2] combined LTL-based RV monitors with MBD, using detected violations to compute minimal sets of abnormal components. However, monitoring and diagnosis remain largely separate: RV analyzes system behavior over potentially unbounded state sequences, whereas diagnosis is performed at a single, unspecified point in time after faults have been detected.

Köhl et al. [20] propose event-based diagnosis using a single model for both normative and failure actions, enabling simultaneous monitoring and diagnosis. While this identifies which failure actions occurred, it does not directly map them to abnormal components in the MBD sense.

A prominent line of work in RV is *stream-based RV*, modeling monitoring as synchronous transformations of input streams to output streams via equational specifications. This paradigm, pioneered in LOLA [8], has recently been extended to handle uncertain inputs and environmental assumptions, allowing refined inference based on expert knowledge [14]. Observations are unfolded on the fly and evaluated using an SMT solver to produce verdicts at each time point.

In this paper, we integrate stream-based RV with MBD based on LOLA, yielding a sound, unified framework for online diagnosis. Our contributions are:

1. **Unified Encoding.** System descriptions are expressed as specialized streams which implicitly represent all diagnoses for observed system behavior. Constraints generated and maintained in LOLA are combined with an SMT solver and standard diagnosis algorithms to derive minimal diagnoses at runtime.
2. **Multi-Instant Diagnosis.** To explain time-invariant faults, e.g., when the observation horizon is negligible compared to typical component lifetimes, we propose a theory dubbed *multi-instant diagnosis*, which subsumes the well-known *diagnosis over multiple observations* [16] as a special case.
3. **Temporal Diagnosis.** By accounting for time-varying abnormality, our framework supports *temporal diagnosis* [12], identifying not only faulty components but also when faults occur. This is critical for systems with transient failures, e.g., overheated components or intermittently failing network links.
4. **Various Other Diagnostic Scenarios.** We discuss further diagnostic settings that can be elegantly addressed in the LOLA RV framework—e.g., background knowledge, abstract modeling, as well as fault models and types.
5. **Implementation and Evaluation.** We prototypically implemented the framework and validated it on two circuits from ISCAS85 [5] and ISCAS89 [4] benchmarks, showing its feasibility and providing first performance metrics.

Overall, our work provides a compact, well-founded integration of RV and MBD, supporting a multitude of diagnostic use cases in a single, stream-based framework, bridging theoretical rigor with practical runtime applicability.

2 Preliminaries

2.1 Model-Based Diagnosis (MBD)

We briefly review the general MBD framework [26], based on [36]. Let \mathcal{L} be a decidable monotonic knowledge representation language. Throughout this paper, all formulas are assumed to be over some fixed \mathcal{L} .

Diagnosed System. The subject of a diagnosis task is a *system*, e.g., a digital circuit or physical device. Formally, a system is a tuple $(SD, COMPS)$ where SD , the system description, is a set of formulas, and $COMPS$, the system components, is a finite set of constants. The distinguished “abnormal” predicate AB is used in

SD to model the expected behavior beh_c of components $c \in \text{COMPS}$. So, SD can be split into two partitions, one, SD_{beh} , that includes all formulas in which the AB predicate appears (*behavioral knowledge*), and the other, SD_{gen} , including all remaining formulas (*general knowledge*). Let, for now, $SD_{beh} := \{\neg \text{AB}(c) \rightarrow beh_c \mid c \in \text{COMPS}\}$, which subsumes one statement of the form “if c is normal, then its behavior is beh_c ” for each $c \in \text{COMPS}$. In contrast, SD_{gen} can comprise, e.g., formulas describing the system domain or the interplay between components.

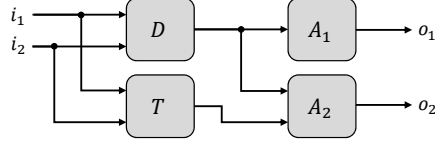


Fig. 1: Running Example

Example 1. Consider an alarm system for a medicine cooling room equipped with two cooling units and two temperature sensors, each located next to a cooling unit. To identify potential failures of the cooling or the sensors, the system with components $\text{COMPS} = \{D, T, A_1, A_2\}$ shown in Fig. 1 is used. The system inputs are the numerical readings i_1, i_2 from the two temperature sensors. D computes the deviation between the sensor readings, and A_1 raises an alarm if this deviation reaches a safety threshold d_{\max} (e.g., indicating a malfunction of one cooling unit and prompting inspection). T computes the average room temperature, and A_2 raises an alarm if both sensors agree on a critical temperature of at least h_{\max} (e.g., suggesting failure of both cooling units or an improperly closed door). Formally, $beh_D := out(D) = |i_1 - i_2|$, $beh_T := out(T) = (i_1 + i_2)/2$, $beh_{A_1} := out(A_1) = (out(D) \geq d_{\max})$, and $beh_{A_2} := out(A_2) = (out(T) - out(D) \geq h_{\max})$. Together, these define SD_{beh} . The system outputs are $o_1 = out(A_1)$ and $o_2 = out(A_2)$, which are part of SD_{gen} .

Diagnosis Problem. The system description can model different behaviors depending on the components’ states. Let $SD^*[\Delta] := SD \cup \{\text{AB}(c) \mid c \in \Delta\} \cup \{\neg \text{AB}(c) \mid c \in \text{COMPS} \setminus \Delta\}$ for $\Delta \subseteq \text{COMPS}$ denote the behavior description of the system (SD, COMPS) under the assumption that components in Δ are abnormal and all others are normal. The *expected behavior* of a correctly working system is thus given by $SD^*[\emptyset]$. The core idea of MBD is to contrast this expected behavior with *observations* of the actual system behavior, represented by a finite set of formulas OBS (e.g., encoding inputs and outputs). A system together with observations constitutes a *diagnosis problem instance (DPI)* (SD, COMPS, OBS).

Fault Detection and Localization. Testing whether the expected behavior contradicts the observations, formally $SD^*[\emptyset] \cup \text{OBS}$ being inconsistent, is called *fault detection*. If positive, the goal of *fault localization* is to identify the components causing the misbehavior. To this end, MBD employs the notion of a *diagnosis*, i.e., an assumption about the state (normal or abnormal) of each component that would explain the discrepancy between expected and observed behavior. Formally, $\Delta \subseteq \text{COMPS}$ is a diagnosis for DPI (SD, COMPS, OBS) iff $SD^*[\Delta] \cup \text{OBS}$ is consistent; it is *minimal* iff there is no diagnosis $\Delta' \subset \Delta$.

Given SD_{beh} as defined above, only the normal behavior of each component is specified. This is referred to as *weak-fault model (WFM)* [33], and our assumption throughout this work. Using a WFM, minimal diagnoses are representative of all

diagnoses, i.e., a set is a diagnosis iff it is a superset of some minimal diagnosis. Hence, fault localization in this case focuses only on minimal diagnoses.

Diagnosis Computation. Basically, the size of the search space for minimal diagnoses is in $\mathcal{O}(2^{|\text{COMPS}|})$. A key concept to restrict this search space is that of a *conflict* [26,19], i.e., a set of components that cannot all be normal given OBS. Formally, $C \subseteq \text{COMPS}$ is a *conflict* for a DPI (SD, COMPS, OBS) iff $\text{SD} \cup \text{OBS} \cup \{\neg \text{AB}(c) \mid c \in C\}$ is inconsistent; it is *minimal* iff there is no conflict $C' \subset C$.

Clearly, any explanation of the system misbehavior must assume at least one component in each minimal conflict to be abnormal. In other words [26]: *Every (minimal) diagnosis for a DPI is a (minimal) hitting set of all minimal conflicts for this DPI.* A (minimal) hitting set of a collection $S = \{S_1, \dots, S_n\}$ is a (\subseteq -minimal) set H such that $H \subseteq \bigcup_{S_i \in S} S_i$ and $H \cap S_i \neq \emptyset$ for all $i = 1, \dots, n$. Hence, a common way of diagnosis computation is by first identifying minimal conflicts (e.g., using [17,29]) and then calculating their minimal hitting sets.

Various diagnosis computation techniques are available for fault localization, including hitting-set-based best-first approaches [26,13,30,32], knowledge compilation methods [9,39,24], stochastic strategies [11,37], and direct techniques [38,31]. A comprehensive overview is provided in [33].

Example 2. Recall the alarm system from Ex. 1, and let $d_{\max} := 1$ and $h_{\max} := 5$. Assume we observe its inputs and outputs as $(i_1, i_2, o_1, o_2) = (5, 7, \text{false}, \text{true})$. First, a *fault detection* is executed: Assuming all components to be normal, the predicted system behavior is i.a. $\text{out}(D) = 2$ (using i_1, i_2) and $o_1 = \text{out}(A_1) = \text{true}$ (using $\text{out}(D)$), which contradicts the observation $o_1 = \text{false}$. Hence, there must be some fault in the system. For *fault localization*, conflict computation first yields two *minimal conflicts* $C_1 = \{D, A_1\}$ (see derivation above) and $C_2 = \{D, T, A_2\}$ (because assuming these three components to be normal, $o_2 = \text{false}$ can be inferred). Second, the computation of minimal hitting sets of $\{C_1, C_2\}$ yields the *minimal diagnoses* $\Delta_1 = \{D\}$, $\Delta_2 = \{A_1, T\}$, and $\Delta_3 = \{A_1, A_2\}$.

Multiple Diagnoses. In practice, many minimal diagnoses may exist, and computing them is NP-hard [6]. As distinct diagnoses encode mutually exclusive assumptions, the goal is to identify the *actual diagnosis*, i.e., the faulty components, and dismiss *spurious* ones. This is addressed via *ranking* (ordering diagnoses, e.g., by minimal cardinality or maximal probability), *filtering* (computing only top-ranked diagnoses), and *discrimination* (acquiring observations to prune the diagnosis space). The latter defines the *sequential diagnosis problem* [19,34]: Given DPI = (SD, COMPS, OBS), find observations OBS_{new} such that a single (highly probable) minimal diagnosis remains for (SD, COMPS, OBS \cup OBS_{new}).

Multiple Observations. In the classic MBD setting, one considers a single observation at one time point. In scenarios such as RV, however, the system can be observed multiple times (possibly under different inputs or revealing previously unobserved values). Depending on whether component (ab)normality is assumed *static* or *dynamic* across observations, this yields a *multi-observation (MO) diagnosis* or a *temporal diagnosis* setting. The MO setting is relevant in systems like circuits, where component lifespans far exceed the observation

period, while the temporal setting is useful for problems with frequent faults or long observation periods, e.g., networks with intermittently failing routers.

In MO diagnosis, given $\text{OBS}_1, \dots, \text{OBS}_k$, the goal is to find assumptions about component (ab)normality consistent with *all* observations. Formally, $\Delta \subseteq \text{COMPS}$ is a (*minimal*) *MO-diagnosis* iff $\text{SD}^*[\Delta] \cup \text{OBS}_i$ is consistent for all $i = 1, \dots, k$ (and no strict subset satisfies this). Existing methods [18,16] either encode and solve directly, or solve the k single-observation DPis and merge results. The latter uses *minimal aggregated diagnoses*, which coincide with minimal MO-diagnoses: $\Delta \subseteq \text{COMPS}$ is such a diagnosis [16] iff $\Delta = \bigcup_{i=1}^k \Delta_i$, where $\Delta_i \in \mathbf{D}_i$ and \mathbf{D}_i is the set of minimal diagnoses for OBS_i , and no strict subset satisfies this.

In temporal diagnosis, component states may change over time. Given time-stamped observations $(\text{OBS}_1, t_1), \dots, (\text{OBS}_k, t_k)$, the goal is to find assumptions about component (ab)normality *at each observation time point* consistent with all observations. Formally, a *temporal diagnosis* [12] is a set $\Delta_t \subset \text{COMPS} \times \{t_1, \dots, t_k\}$, where $(c, t_i) \in \Delta_t$ denotes $\text{AB}(c)$ at t_i and $(c, t_i) \notin \Delta_t$ means $\neg \text{AB}(c)$ at t_i , such that $\text{SD} \cup \bigcup_{i=1}^k (\text{OBS}_i, t_i) \cup \Delta_t$ is consistent.

2.2 Symbolic Runtime Verification (RV) using LOLA

We briefly summarize symbolic RV with LOLA based on [14].

LOLA Syntax. A (*synchronous*) *stream* is a function $s: \mathbb{T} \rightarrow \mathbb{D}$ assigning each instant in a time domain \mathbb{T} a value of a type \mathbb{D} (e.g., Boolean \mathbb{B} , real \mathbb{R}), where \mathbb{T} is infinite (\mathbb{N}) or finite ($\{0, 1, \dots, t_{max}\}$). We write streams as sequences, e.g., $s: \{0, 1, 2\} \rightarrow \mathbb{R}$ with $s(0) = 4, s(1) = 3, s(2) = 2$ is written as $s = \langle 4, 3, 2 \rangle$.

A *LOLA specification* transforms *input streams* to defined streams [8]. The values of *defined streams* are obtained by evaluating expressions that depend on the values of (other) streams. Furthermore, a special Boolean *assumption stream* holds by definition, allowing background knowledge to be encoded and thereby enabling more precise reasoning. Finally, *internal streams* are neither given as inputs nor defined by an expression, but implicitly defined via the assumption stream. Formally, a LOLA specification is a tuple $\varphi = (I, D, J, E, \alpha)$ consisting of pairwise disjoint sets I, D, J of typed identifiers for input streams, defined streams and internal streams, respectively; a distinguished assumption stream identifier α of type \mathbb{B} ; and a function E assigning to every $y \in D \cup \{\alpha\}$ of type \mathbb{D} a defining *LOLA expression* E_y of type \mathbb{D} .

The *set of all LOLA expressions* of type \mathbb{D} is denoted by $\mathbb{E}_{\mathbb{D}}$ and is recursively defined as $\mathbb{E}_{\mathbb{D}} := c \mid s[o|c] \mid f(e_1, \dots, e_n)$ where c is a constant of type \mathbb{D} , $s \in I \cup D \cup J$ is a stream identifier of type \mathbb{D} , $o \in \mathbb{Z}$ is an offset, f is a function symbol mapping from types $\mathbb{D}_1, \dots, \mathbb{D}_n$ to \mathbb{D} , and $e_i \in \mathbb{E}_{\mathbb{D}_i}$. The offset operator $s[o|c]$ represents the stream that has at time t the value of stream s at time $t+o$, and the value c if $t+o \notin \mathbb{T}$. In case of $o = 0$, we simply write s . A LOLA specification is called *atemporal* iff every offset $o = 0$. An input stream can be *certain*, i.e., a deterministic sequence of constants, or *uncertain*, i.e., a (partially) nondeterministic sequence of values, e.g., due to noisy sensors, missing observations, or other sources of inaccuracy. We call a (possibly nondeterministic) sequence of values for a set of streams over a sequence of time instants a *trace*.

```

in  ld : ℝ
def  acc := acc[-1|0] + ld - ld[-3|0]
def  ok  := acc ≤ 8
α := 0 ≤ ld ≤ 5

```

Listing 1: Simple LOLA specification

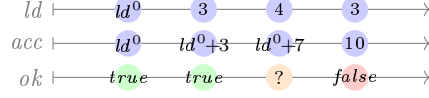


Fig. 2: Trace for Listing 1

Example 3. Assume a LOLA specification $\varphi = (\{ld\}, \{acc, ok\}, \emptyset, E, \alpha)$ with E_y for $y \in \{acc, ok, \alpha\}$ as defined in Listing 1, and a corresponding trace shown in Fig. 2. The defined stream acc accumulates CPU load measurements from the input stream ld by adding the current load to the previous accumulated value and subtracting the third-last input value. Note, $acc[-1|0]$ refers to the value acc attained one time instant earlier; it starts with 0 at $t = 0$. The defined stream ok checks if the current acc is at most 8. Due to α , ld is known to be always in $[0, 5]$. Moreover, ld is an uncertain input stream as its first value is unknown.

LOLA Semantics. In case all input streams are certain, the LOLA semantics deterministically maps all input streams to unique defined streams [8]. Given uncertain input streams, in contrast, there are generally multiple possible values input, defined, and internal streams can attain at each given time instant. To represent the possible values of a stream $s \in I \cup D \cup J$ at an instant $t \in \mathbb{T}$, the concept of an *instant variable* s^t is used (cf. ld^0 in Fig. 2).

Even though nondeterministic, values of uncertain input streams might nevertheless be constrained. To capture this, LOLA allows to specify *input constraints*, (in)equalities over input stream instant variables specifying all possible input values at a given time. More specifically, given an input stream identifier $x \in I$, a constant c can be represented as constraint $x^t = c$ (deterministic case), a closed interval $[l, r]$ as $l \leq x^t \leq r$ (analogously for open intervals), and a finite set A as $\bigvee_{a \in A} x^t = a$. An entirely unknown input is simply represented by not specifying any input constraint for x^t . The *set of all input constraints at time t* is denoted by ψ^t (e.g., $\psi^0 = \emptyset$ and $\psi^1 = \{ld^1 = 3\}$ in Fig. 2).

Since input streams are represented by constrained instant variables instead of fixed values, the values of defined streams depend on these variables. This dependence is captured using *symbolic expressions*, i.e., algebraic expressions over instant variables. For a type \mathbb{D} , we write $Expr_{\mathbb{D}}$ for the set of all symbolic expressions of that type. The symbolic expression for each defined stream $y \in D$ is characterized by its LOLA expression E_y . Hence, for each instant $t \in \mathbb{T}$, this yields a constraint of the form $y^t = [E_y]_{sym}(t)$. Here, the function $[e]_{sym} : \mathbb{T} \rightarrow Expr_{\mathbb{D}}$ returns the *symbolic instantiation* of a LOLA expression $e \in \mathbb{E}_{\mathbb{D}}$ at time t (e.g., for $t = 2$ in Listing 1, $X := \{acc^2 = acc^1 + ld^2 - 0, ok^2 = acc^2 \leq 8\}$ includes the constraints obtained in this way for acc and ok). The symbolic instantiation is recursively defined as $[c]_{sym}(t) = c$ for a constant c , $[f(e_1, \dots, e_n)]_{sym}(t) = f([e_1]_{sym}(t), \dots, [e_n]_{sym}(t))$, $[s[o|c]]_{sym}(t) = s^{t+o}$ if $t+o \in \mathbb{T}$, and $[s[o|c]]_{sym}(t) = c$ if $t+o \notin \mathbb{T}$. Similarly as for defined streams, the LOLA expression E_{α} assigned to the assumption stream α induces a constraint for every instant $t \in \mathbb{T}$ by

computing the symbolic instantiation $[E_\alpha]_{sym}(t)$ of E_α (e.g., for instant 2 in Listing 1, $Y := \{0 \leq ld^2 \leq 5\}$ includes the constraint derived from stream α).

Putting together the constraint generation for defined streams and for the assumption stream, we obtain what is commonly referred to as the *instantiation* $[\varphi]_{sym}(t) = \{y^t = [E_y]_{sym}(t) \mid y \in D\} \cup \{[E_\alpha]_{sym}(t)\}$ of a LOLA specification $\varphi = (I, D, J, E, \alpha)$ for a time instant $t \in \mathbb{T}$ (e.g., $[\varphi]_{sym}(2)$ for the LOLA specification φ given in Ex. 3 is equal to $X \cup Y$ for X and Y discussed above).

Now, by combining the specification's instantiation with the input constraints over all time instants, formally $\bigcup_{t \in \mathbb{T}} [\varphi]_{sym}(t) \cup \psi^t$, all available information is captured. Based on this, the LOLA semantics specifies any combination of values of instant variables as a possible world (a *model*) if it satisfies all these constraints, and as impossible otherwise (e.g., in Ex. 3, the tuple of streams $(\langle 1, 3, 4, 3 \rangle, \langle 1, 7, 11, 10 \rangle, \langle true, true, true, false \rangle)$ for time instants $0 \leq t \leq 3$ corresponding to the stream identifiers (ld, acc, ok) represents a model). We then say that some symbolic expression $e \in Expr_{\mathbb{D}}$ is entailed, formally $\bigcup_{t \in \mathbb{T}} [\varphi]_{sym}(t) \cup \psi^t \models e$, iff e is true in all models (e.g., $acc^3 = 10$ is an entailment in Ex. 3).

Example 4. Recall the LOLA specification φ from Ex. 3. Despite the entirely unknown value of ld at instant $t = 0$ (no input constraint for ld^0), the constraint set $\bigcup_{t \in \mathbb{T}} [\varphi]_{sym}(t) \cup \psi^t$ allows to infer that $ld^0 \in [0, 5]$ and thus ok^0 and ok^1 are *true*. For $t = 2$, however, $acc^2 \in [7, 12]$ is entailed, which is why ok^2 is unknown, i.e., neither *true* nor *false* follows for ok^2 . At $t = 3$, the constraint $acc^3 = acc^2 + ld^3 - ld^0$ involves a subtraction of (the only nondeterministic value) ld^0 . Hence, the deterministic values $acc^3 = 10$ and $ok^3 = false$ are implied.

Symbolic Stream Reasoning in LOLA. In online monitoring, inputs are only known up to the current instant t ; future inputs are unknown. In order to perform symbolic reasoning using the current information (as shown in Ex. 4), the accumulated constraint set $[\varphi]_{\bar{\psi}}^{\leq t} := \bigcup_{t'=0}^t ([\varphi]_{sym}(t') \cup \psi^{t'})$ needs to be stored and managed. A program that incrementally maintains and updates $[\varphi]_{\bar{\psi}}^{\leq t}$ is called a *symbolic monitor* M , whose state at time t is $M^t = [\varphi]_{\bar{\psi}}^{\leq t}$. When inputs for $t + 1$ become available, M^{t+1} is built by adding $[\varphi]_{sym}(t + 1)$ and ψ^{t+1} to M^t . Using M^t , one can reason about the system up to time t by means of an SMT solver (e.g., Z3 [25]), e.g., to determine the possible values of a stream.

Since new constraints are added at every instant, M^t grows over time, leading to increasing runtime and memory consumption. In many applications, however, monitoring is required to run in *constant* time and space, independent of t . To achieve this, *pruning strategies* [14] rewrite constraints so that the size of M^t remains bounded. A symbolic monitor is *sound* iff, after pruning, it allows to deduce at least the same values for a set of instant variables of interest as before pruning. It is *perfect* iff it deduces exactly the same values. A sound but imperfect monitor may overapproximate, i.e., allow additional values, which can be necessary to guarantee constant memory usage [14].

3 Combining RV and MBD using LOLA

In this section, our goal is to combine RV with MBD in the context of LOLA. This allows for a continuous observation of system executions, and for the reasoning about possible component faults, and thus synergizes the strengths of RV in symbolic stream-based reasoning and efficient fault detection with the merits of MBD as a powerful framework for fault localization.

In Sec. 3.1, we expand on how and which diagnosis problems can be specified using LOLA. We then demonstrate system diagnosis in the stream-reasoning scenario. Sec. 3.2 considers component fault states assumed *static* over the observation horizon and develops a generalized multi-observation diagnosis theory (cf. Sec. 2.1). Sec. 3.3 focuses on *dynamic* component fault states and the associated temporal diagnosis scenario in the RV context (cf. Sec. 2.1). Sec. 3.4 demonstrates a proof-of-concept implementation. Finally, Sec. 3.5 briefly discusses additional diagnostic settings that can be elegantly represented in the LOLA framework—e.g., background knowledge, monitors, abstract system modeling, fault models, and different fault types with respect to their evolution over time. All proofs for the lemmas and theorems are included in Appendix A.

3.1 Specifying Diagnosis Problems in LOLA

In what follows, we consider diagnosed systems $S = (SD, COMPS)$ as per Sec. 2.1 which are *synchronous* (all variables update simultaneously at each time step), *discrete-time* (time progresses in steps, not continuously), and *expressible in LOLA*. Examples of system types that can be eligible are combinatorial and sequential circuits [23], spreadsheets [35], software [3], (cyber)physical systems [1], and discrete-event systems [7].

To map S to a LOLA specification φ , we have to decide how the system description SD and the system components $COMPS$ as well as the system variables (e.g., inputs, outputs, and internal values) can be captured by φ . Aligning with the syntax and semantics of LOLA expounded in Sec. 2.2, the basic idea is to

- specify SD in terms of the assumption stream, which is true by definition,
- model all observable (but not necessarily observed) system aspects of interest in terms of (uncertain) input streams, e.g., (noisy or intermittent) sensor information about system inputs, outputs or internal values, and
- define all unobservable system aspects we might be interested in inferring in terms of internal streams, e.g., the component (ab)normality states or internal system values not measurable or monitored by a sensor.

Defined streams, although not required to describe S , can be exploited, e.g., to specify auxiliary streams for system modeling, or to define monitors [22] that guard certain system aspects of interest, e.g., necessary invariants or integrity criteria. In this vein, if yielding a verdict *false*, such monitors can serve as useful indicators of system malfunctions that trigger fault localization actions, as perpetual diagnostic reasoning might be infeasible due to system size or complexity.

Definition 1 (LOLA System). *Let $S = (SD, COMPS)$ be a system as per Sec. 2.1 which is synchronous, discrete-time, and expressible in LOLA. Let $V_o(SD)$ (and*

```

internal  $D, T, A_1, A_2 : \mathbb{B}$ 
in  $i_1, i_2 : \mathbb{R}$     in  $out_{A_1}, out_{A_2} : \mathbb{B}$ 
internal  $out_D, out_T : \mathbb{R}$ 
def  $mon := out_{A_2} \rightarrow \min(i_1, i_2) \geq 5$ 
 $\alpha := (\neg D \rightarrow (out_D = |i_1 - i_2|))$ 
     $\wedge (\neg T \rightarrow (out_T = (i_1 + i_2)/2))$ 
     $\wedge (\neg A_1 \rightarrow (out_{A_1} \leftrightarrow out_D \geq 1))$ 
     $\wedge (\neg A_2 \rightarrow (out_{A_2} \leftrightarrow out_T - out_D \geq 5))$ 
    
```

Listing 2: LOLA specification

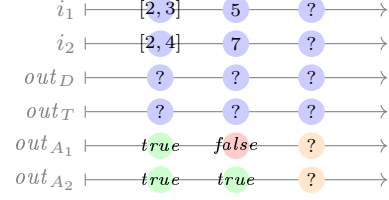


Fig. 3: Trace for Listings 2 and 3

$V_u(SD)$ be functions that extract all observable (unobservable) variables from the system description SD . Then, the pair $(\varphi, COMPS)$ is a LOLA system for S iff $\varphi = (I, D, J, E, \alpha)$ is a LOLA specification where

- the assumption stream $\alpha := SD$ (system description),
- there is an internal stream $c \in J$ of type \mathbb{B} for each $c \in COMPS$ where a value *true* indicates that component c is abnormal (component health states),
- there is an input stream $o \in I$ of type \mathbb{D} for each $o \in V_o(SD)$ of type \mathbb{D} (observable system variables), and
- there is an internal stream $u \in J$ of type \mathbb{D} for each $u \in V_u(SD)$ of type \mathbb{D} (unobservable system variables).

A LOLA system represents a mere description of the system to be diagnosed in terms of a LOLA specification. What is still missing are observations about the system. These are obtained by the input constraints $\Psi = \bigcup_{t'=0}^t \psi^{t'}$ up to the current time instant t , because only these give (possibly uncertain) information about the actual input streams. Together with the LOLA system, this yields the full information needed to specify a LOLA diagnosis problem instance. Note that, in case of atemporal systems, where only 0-offsets are used, all time instants can be considered separately and independently; i.e., only ψ^t , instead of the entire input constraint history, is relevant to the diagnostic reasoning at instant t .

Definition 2 (LOLA DPI). Let $(\varphi, COMPS)$ with $\varphi = (I, D, J, E, \alpha)$ be a LOLA system and $\Psi = \bigcup_{t'=0}^t \psi^{t'}$ be the set of input constraints for all streams $s \in I$ for the instants $0, \dots, t$. Then, the tuple $(\varphi, COMPS, \Psi)$ is called a LOLA diagnosis problem instance (LOLA DPI).

Example 5. Recall our alarm system from Fig. 1, described in Examples 1 and 2. Assume that the system's inputs (i_1, i_2) and outputs (out_{A_1}, out_{A_2}) have attached (noisy) sensors and are thus observable, and let the internal variables (out_D, out_T) be unobservable. Then, a LOLA DPI as per Def. 2 corresponding to this system is given by $(\varphi, COMPS, \Psi)$ where $COMPS = \{D, T, A_1, A_2\}$, φ is given in Listing 2, and $\Psi = \{\psi^0, \psi^1, \psi^2\}$ is shown by the two top and the two bottom traces in Fig. 3. Note: (1) The sensors for the system inputs yield an output that is nondeterministic at instant 0, deterministic at 1, and entirely unknown at 2, whereas the sensors readings for the output values are deterministic at 0

and 1, but entirely unknown at 2 (cf. Fig. 3). (2) The defined stream *mon* is not part of the LOLA DPI, but constitutes a monitor of the integrity criterion that $\min(i_1, i_2) \geq 5$ must hold whenever *out*_{A₂} is *true*. Thus, if the value of *mon* is *false*, *fault detection* is positive, and *fault localization* can be started, involving reasoning about possible sets of faulty components (values of the internal streams in COMPS) using the symbolic monitor for φ and Ψ (cf. Sec. 2.2) along with an SMT solver and off-the-shelf MBD algorithms (cf. Sec. 2.1).

3.2 Multi-Instant Diagnosis with LOLA

Let us first consider the common setting where component states are static over time. The basic idea underlying our diagnosis approach is that the symbolic monitor maintains a consistent constraint set $[\varphi]_{\Psi}^{\leq t}$ for current instant t (see Sec. 2.2), from which diagnoses can be inferred. Specifically, a diagnosis Δ encodes an assumption about each component's state, i.e., normal or abnormal, which is consistent with $[\varphi]_{\Psi}^{\leq t}$. A component c at instant t' assumed to be normal ($c \notin \Delta$) means that $\neg c^{t'}$ holds, and if assumed to be abnormal ($c \in \Delta$) $c^{t'}$ holds. The question is for which instants t' each component's state should be assumed. Two natural options are to only consider the current instant ($t' = t$) or additionally also the entire past ($t' \leq t$). The former does not make assumptions about past component behavior, while the latter assumes the same states throughout the past. To generalize, we consider diagnoses conforming to k past instants. We denote by $t \downarrow k := \{t' \in \mathbb{T} \mid t - k \leq t' \leq t\}$ all instants from $t - k$ until t .

Definition 3 (k -Instant Diagnosis). *Given a LOLA DPI $(\varphi, \text{COMPS}, \Psi)$ and $k \in \mathbb{N}$, $\Delta \subseteq \text{COMPS}$ is a k -instant diagnosis for $t \in \mathbb{T}$ iff $\{c^{t'} \mid c \in \Delta \wedge t' \in t \downarrow k\} \cup \{\neg c^{t'} \mid c \in \text{COMPS} \setminus \Delta \wedge t' \in t \downarrow k\} \cup [\varphi]_{\Psi}^{\leq t}$ is consistent. A k -instant diagnosis Δ is minimal iff no $\Delta' \subset \Delta$ is a k -instant diagnosis.*

Intuitively, 0-instant diagnosis is akin to diagnosis with a single observation (the current inputs), and k -instant diagnosis with $k > 0$ is similar to multi-observation diagnosis over k observations [16]. In fact, as we will see later on, this connection holds up as long as φ is atemporal, i.e., without non-zero offsets.

Now, an obvious question is why one should not always choose a large value for k . The drawback is that a pruning strategy (cf. Sec. 2.2) must retain variables from the past k instants; only variables older than $t - k$ can be pruned. This increases the symbolic monitor's size. Nevertheless, as long as k is constant, a constant pruning strategy guarantees constant-time diagnosis, in the sense that complexity is independent of t . Particularly, constant monitoring is impossible if the entire past up to t ($k = t$) is considered. However, when a fault is detected, it might be a sensible assumption that it is likely to have occurred rather recently.

To get an intuition, let us first consider 0-instant diagnosis for atemporal φ .

Example 6. Recall Ex. 5. We assume inputs for instants $t = 0, 1, 2$ as given by Fig. 3. The minimal 0-instant diagnoses per t are shown in Tab. 1.

At instant $t = 0$, we have $[\varphi]_{\text{sym}}(0) = \{(\neg D^0 \rightarrow \text{out}_D^0 = |i_1^0 - i_2^0|) \wedge (\neg T^0 \rightarrow \text{out}_T^0 = (i_1^0 + i_2^0)/2) \wedge \dots\}$. The sensors for the inputs deliver noisy values reflected

	$t = 0$	$t = 1$	$t = 2$
0-instant	$\{D, A_1\}, \{T\}, \{A_2\}$	$\{D\}, \{A_1, A_2\}, \{A_1, T\}$	\emptyset
1-instant	$\{D, A_1\}, \{T\}, \{A_2\}$	$\{D, A_1\}, \{D, T\}, \{D, A_2\},$ $\{A_1, T\}, \{A_1, A_2\}$	$\{D\}, \{A_1, A_2\}, \{A_1, T\}$
2-instant/ aggr of 0's	$\{D, A_1\}, \{T\}, \{A_2\}$	$\{D, A_1\}, \{D, T\}, \{D, A_2\},$ $\{A_1, T\}, \{A_1, A_2\}$	$\{D, A_1\}, \{D, T\}, \{D, A_2\},$ $\{A_1, T\}, \{A_1, A_2\}$

Table 1: Minimal diagnoses for trace from Fig. 3 on specification from Listing 2. Row 3 also shows minimal aggregations of all 0-instant diagnoses for $0, \dots, t$.

by constraints $\psi^0 = \{2 \leq i_1^0 \leq 3, 2 \leq i_2^0 \leq 4, out_{A_1}^0 = true, out_{A_2}^0 = true\}$. Thus, $[\varphi]_{\bar{\psi}}^0 = [\varphi]_{sym}(0) \cup \psi^0$. Since there is only evidence for $out_{A_2}^0$ being incorrect ($out_T^0 - out_D^0 = [2, 3.5] - [0, 2] = [0, 3.5] \not\geq 5$), the obvious conflict is $\{D, A_2, T\}$, leading to diagnoses $\{D\}, \{A_2\}, \{T\}$ (for A_1 we have $out_D^0 \in [0, 2]$ which does not imply a fault). However, $\{D\}$ is actually not valid due to $D^0 \wedge \neg T^0 \wedge \neg A_1^0 \wedge \neg A_2^0$ being inconsistent with $[\varphi]_{\bar{\psi}}^0$: $[2, 3.5] - out_D^0 \geq 5$ implies out_D^0 being negative and A_1 therefore being abnormal, too (because the normal behavior of D can only produce non-negatives).

For $t = 1$, $[\varphi]_{sym}(1)$ and $\psi^1 = \{i_1^1 = 5, i_2^1 = 7, out_{A_1}^1 = false, out_{A_2}^1 = true\}$ are added, and diagnostic reasoning is identical to Ex. 2. For $t = 2$, since i_1^2 and i_2^2 are fully unknown, any assignment for the outputs out_{A_1} and out_{A_2} is consistent with $[\varphi]_{\bar{\psi}}^2$, leading to \emptyset being the only minimal diagnosis.

Similar to how a multi-observation diagnosis is also a diagnosis for each single observation [16], since a k -instant diagnosis makes assumptions about each component's state over the past k instants, it is also a 0-instant diagnosis for the past k instants. Particularly, if k is set to the current instant t , it explains all observations for instants $0, \dots, t$. Tab. 1 depicts the minimal k -instant diagnoses for $k = 1$ and $k = 2$. Since 1-instant diagnosis no longer reasons at $t = 2$ about the component states for instant 0, it cannot explain faults for that instant.

Lemma 1. *Let $(\varphi, COMPS, \Psi)$ be a LOLA DPI. Given $t \in \mathbb{T}$ and $k \in \mathbb{N}$, every k -instant diagnosis for t is also a 0-instant diagnosis for every $t' \in t \downarrow k$.*

As mentioned in Sec. 2.1, a common technique to obtain diagnoses for multiple observations is to compute the diagnoses for each observation separately and to aggregate them afterwards. In fact, this also works with k -instant diagnosis if φ is atemporal (Theorem 1 (ii)). Let us denote by $aggr(\mathbf{D}_0, \dots, \mathbf{D}_n) = \{\bigcup_{i=0}^n \Delta_i \mid \Delta_i \in \mathbf{D}_i\}$ the set of aggregated diagnoses over sets of diagnoses \mathbf{D}_i . E.g., row 3 of Tab. 1 depicts for each t the minimal aggregated 0-instant diagnoses for $0, \dots, t$; they are identical to the 2-instant diagnoses. However, if φ involves non-zero offsets, i.e., temporal dependencies, this is not guaranteed:

Example 7. Consider the diagnosis specification φ from Listing 3 which is an extension of Listing 2, where D is no longer specified to compute the deviation between the current i_1 and i_2 but to keep track of the sum of the three most

```

internal  $D, T, A_1, A_2 : \mathbb{B}$            internal  $out_D, out_T : \mathbb{R}$ 
in  $i_1, i_2 : \mathbb{R}$                        in  $out_{A_1}, out_{A_2} : \mathbb{B}$ 
def  $dev := |i_1 - i_2|$                  def  $avg := (i_1 + i_2)/2$ 
 $\alpha := (\neg D \rightarrow (out_D = out_D[-1, 0] + dev - dev[-3, 0]))$ 
       $\wedge (\neg T \rightarrow (out_T = out_T[-1, 0] + avg - avg[-3, 0]))$ 
       $\wedge (\neg A_1 \rightarrow (out_{A_1} \leftrightarrow out_D \geq 1)) \wedge (\neg A_2 \rightarrow (out_{A_2} \leftrightarrow out_T - out_D \geq 5))$ 

```

Listing 3: Temporal extension of Listing 2.

	$t = 0$	$t = 1$	$t = 2$
0-instant	$\{D, A_1\}, \{T\}, \{A_2\}$	\emptyset	\emptyset
aggr of 0's	$\{D, A_1\}, \{T\}, \{A_2\}$	$\{D, A_1\}, \{T\}, \{A_2\}$	$\{D, A_1\}, \{T\}, \{A_2\}$
1-instant	$\{D, A_1\}, \{T\}, \{A_2\}$	$\{D, A_1\}, \{D, T\}, \{D, A_2\},$ $\{A_1, T\}, \{A_1, A_2\}$	\emptyset
2-instant	$\{D, A_1\}, \{T\}, \{A_2\}$	$\{D, A_1\}, \{D, T\}, \{D, A_2\},$ $\{A_1, T\}, \{A_1, A_2\}$	$\{D, A_1\}, \{D, T\}, \{D, A_2\},$ $\{A_1, T\}, \{A_1, A_2\}$
1-temporal	$\{(D, 0), (A_1, 0)\},$ $\{(T, 0)\}, \{(A_2, 0)\}$	$\{(T, 0), (A_1, 1)\}, \{(A_2, 0), (A_1, 1)\},$ $\{(D, 0), (A_1, 0)\}, \{(T, 0), (D, 1)\},$ $\{(A_2, 0), (D, 1)\}$	\emptyset
2-temporal	$\{(D, 0), (A_1, 0)\},$ $\{(T, 0)\}, \{(A_2, 0)\}$	$\{(T, 0), (A_1, 1)\}, \{(A_2, 0), (A_1, 1)\},$ $\{(D, 0), (A_1, 0)\}, \{(T, 0), (D, 1)\},$ $\{(A_2, 0), (D, 1)\}$	$\{(T, 0), (A_1, 1)\}, \{(A_2, 0), (A_1, 1)\},$ $\{(D, 0), (A_1, 0)\}, \{(T, 0), (D, 1)\},$ $\{(A_2, 0), (D, 1)\}$

Table 2: Minimal diagnoses for trace from Fig. 3 on specification from Listing 3. The second row shows minimal aggregations of all 0-instant diagnoses for $0, \dots, t$.

recent deviations (analogously for T and the average). Assuming again inputs from Fig. 3, for $t = 0$, reasoning remains unchanged compared to Ex. 6. However, for $t = 1$, \emptyset is the only minimal 0-instant diagnosis because no values for D^0, T^0, A_1^0, A_2^0 are assumed, i.e., the component states are undefined for the past. Specifically, out_D^1 and out_T^1 could in principle assume any value due to their dependence on the past and, as a result, no component has to be faulty at time 1. The minimal aggregated diagnoses up to $t = 1$ are therefore identical to the 0-instant diagnoses for $t = 0$ (see Tab. 2). However, among these, $\{T\}$ and $\{A_2\}$ are actually not 1-instant diagnoses for $t = 1$ because assuming them for both $t = 0$ and $t = 1$ leads to inconsistency. E.g., if $\{T\}$ would be a diagnosis, $\neg D^0 \wedge \neg D^1 \wedge \neg A_1^0 \wedge \neg A_1^1$ would need to be consistent with $[\varphi]_{\overline{y}}^{\leq 1}$. However, $\neg D^0 \wedge \neg A_1^0$ implies $out_D^0 \in [1, 2]$, hence $\neg D^1$ implies $out_D^1 \in [3, 4]$, and finally $out_{A_1}^1 = true$ due to $\neg A_1^1$. This contradicts the observation that $out_{A_1}^1 = false$. Therefore, either D or A_1 must be abnormal together with T .

This example illustrates that an empty 0-instant diagnosis does not imply that there is no fault at all. Instead, it only means that every potential fault could be explained by abnormal behavior in the past, i.e., there is no evidence for components to be faulty *now*. While in Ex. 7 this leads to empty diagnoses, this is not necessarily the case and depends on the temporal interconnections between components and which of them are observed. As (i) in the following

theorem shows, aggregating 0-instant diagnoses is, however, less “focused” than k -instant diagnosis for $k > 0$ if the specification is not atemporal.

Theorem 1. *Let $(\varphi, \text{COMPS}, \Psi)$ be a LOLA DPI. Given $t \in \mathbb{T}$ and $k \in \mathbb{N}$, let \mathbf{D}_{all} be the set of all k -instant diagnoses and $\mathbf{D}'_{all} = \text{aggr}(\mathbf{D}_{t-k}, \dots, \mathbf{D}_t)$ where $\mathbf{D}_{t'}$ is the set of all 0-instant diagnoses for $t' \in t \downarrow k$. Furthermore, let $\mathbf{D}_{min} \subseteq \mathbf{D}_{all}$ and $\mathbf{D}'_{min} \subseteq \mathbf{D}'_{all}$ be their respective subsets containing only minimal diagnoses. It holds that (i) $\mathbf{D}_{all} \subseteq \mathbf{D}'_{all}$, (ii) $\mathbf{D}_{all} = \mathbf{D}'_{all}$ if φ is atemporal, and (iii) $\forall \Delta \in \mathbf{D}_{min}: \exists \Delta' \in \mathbf{D}'_{min}: \Delta' \subseteq \Delta$.*

Note that (i) of Theorem 1 considers *all* diagnoses, not only minimal ones: A minimal k -instant diagnosis might not be a minimal aggregated diagnosis due to not being a 0-instant diagnosis for all past k instants. However, according to (iii), for each minimal k -instant diagnosis there exists a subset that is a minimal aggregation (compare rows 2 and 4 of Tab. 2). Similarly, larger values for k mean a diagnosis explains more instants at once (compare rows 1, 3, and 4 of Tab. 2):

Theorem 2. *Let $(\varphi, \text{COMPS}, \Psi)$ be a LOLA DPI. Given $t \in \mathbb{T}$ and $k, k' \in \mathbb{N}$ with $k' \leq k \leq t$, let \mathbf{D}_{all} and \mathbf{D}'_{all} be the sets of all k -instant diagnoses and all k' -instant diagnoses for t , respectively. Furthermore, let $\mathbf{D}_{min} \subseteq \mathbf{D}_{all}$ and $\mathbf{D}'_{min} \subseteq \mathbf{D}'_{all}$ be their respective subsets containing only minimal diagnoses. It holds that (i) $\mathbf{D}_{all} \subseteq \mathbf{D}'_{all}$ and (ii) $\forall \Delta \in \mathbf{D}_{min}: \exists \Delta' \in \mathbf{D}'_{min}: \Delta' \subseteq \Delta$.*

3.3 Temporal Diagnosis with LOLA

While above approach allows to accurately diagnose the setting where component states are time-invariant, often it might be that a component is only abnormal for *some* time and recovers afterwards, e.g., due to overheating. This allows faults caused by an abnormal component to manifest themselves at a later time when the component may behave correctly again. In general, component states may change arbitrarily. Therefore, we extend our approach by associating each component with the time of the fault:

Definition 4 (k -Temporal Diagnosis). *Given a LOLA DPI $(\varphi, \text{COMPS}, \Psi)$ and $k \in \mathbb{N}$, a set $\Delta \subseteq \text{COMPS} \times t \downarrow k$ is a k -temporal diagnosis for $t \in \mathbb{T}$ iff $\{c^{t'} \mid (c, t') \in \Delta\} \cup \{\neg c^{t'} \mid (c, t') \in (\text{COMPS} \times t \downarrow k) \setminus \Delta\} \cup [\varphi]_{\Psi}^{\leq t}$ is consistent. A k -temporal diagnosis Δ is minimal iff no $\Delta' \subset \Delta$ is a k -temporal diagnosis.*

Example 8. Consider again Listing 3 with inputs from Fig. 3. Tab. 2 depicts the minimal k -temporal diagnoses for $k = 1, 2$ and instants $t = 0, 1, 2$. For all instants, the k -temporal diagnoses are, barring the timestamps, identical to the k -instant diagnosis from before. However, using the additional time information, we can infer, e.g., from $\{(T, 0), (A_1, 1)\}$ that the observation can be explained when T was previously abnormal and A_1 is abnormal now. Particularly, 2-temporal diagnosis concludes with $\{(D, 0), (A_1, 0)\}$ at $t = 1$ that the fault could be entirely in the past.

As the example shows, temporal diagnosis allows for more precise fault localization by pinpointing the possible times of a fault. In fact, as (i) in the following theorem shows, for each k -temporal diagnosis there exists a k -instant timestamp-free counterpart, and vice versa. We denote the omission of timestamps in temporal diagnoses \mathbf{D} as $tsFree(\mathbf{D}) = \{\{c \mid (c, t') \in \Delta\} \mid \Delta \in \mathbf{D}\}$.

Theorem 3. *Let $(\varphi, COMPS, \Psi)$ be a LOLA DPI. Given $k \leq t$, let \mathbf{D}_{all} be the set of all k -instant diagnoses and \mathbf{D}'_{all} the set of all k -temporal diagnoses for t . Furthermore, let $\mathbf{D}_{min} \subseteq \mathbf{D}_{all}$ and $\mathbf{D}'_{min} \subseteq \mathbf{D}'_{all}$ be their respective subsets containing only minimal diagnoses. It holds that (i) $\mathbf{D}_{all} = tsFree(\mathbf{D}'_{all})$ and (ii) $\mathbf{D}_{min} \subseteq tsFree(\mathbf{D}'_{min})$.*

Note that (i) in Theorem 3 considers *all* diagnoses, not only minimal ones. For minimal sets, (i) generally does not hold: E.g., it is possible that $\mathbf{D}'_{min} = \{\{(X, 0)\}, \{(Y, 0), (X, 1)\}\}$, i.e., $tsFree(\mathbf{D}'_{min}) = \{\{X\}, \{Y, X\}\}$, but $\mathbf{D}_{min} = \{\{X\}\}$. Here, the observation can be explained by the non-minimal temporal diagnosis $\{(X, 0), (X, 1)\} \supset \{(X, 0)\}$, which is why $\{X\}$ is a k -instant diagnosis. While $(X, 0)$ is sufficient and thus minimal, $(X, 1)$ is not and must be accompanied by $(Y, 0)$. The k -instant diagnosis cannot make this distinction. But, as stated by (ii), every minimal k -instant diagnosis is contained in the timestamp-free k -temporal diagnoses.

3.4 Implementation and Evaluation

We prototypically extended the symbolic LOLA tool used in [14,15] with the proposed diagnosis approaches as a proof of concept. We implemented the standard HS-Tree algorithm [26,32] together with QuickXplain [17,29] and Z3 [25] as back-end solver. The tool computes all minimal k -instant diagnoses and k -temporal diagnoses per instant. We evaluated our approach on two Boolean circuits, c17 and s27, from the ISCAS85 [5] and ISCAS89 [4] suites³. c17 has six components and is purely combinational (atemporal), whereas s27 has 14 components and is sequential. We translated both circuits into LOLA and injected stuck-at-*false* faults for one component at a time, to produce erroneous outputs. For c17, the fault is permanent; for s27, the fault occurs at instant 3 and persists thereafter.

For c17, we diagnosed all 32 possible input combinations and their outputs for each injected fault in a single trace (since c17 is atemporal each instant is independent). The actual diagnosis $\{c\}$ was identified as k -instant diagnosis if the fault manifested in least one of the last k instants. Moreover, for faulty instants t' , $\{(c, t') \mid t' \in t \downarrow k\}$ was always a k -temporal diagnosis. For s27, where input order matters, we generated ten random traces of ten instants each to produce outputs for each injected fault. Since faults may propagate over time, both approaches localize the actual fault only if its origin lies within the past

³ The experiments were run on a 64-bit Linux machine with an Intel Core i7-13650U CPU and 32 GB RAM. The tool and experiment data is available at <https://gitlab.isp.uni-luebeck.de/Raik.Hipler/lola-mbd-evaluation>.

k instants. Because s27 is sequential and c is faulty from instant 3 onwards, at time $t \geq 3$ localization of the actual diagnosis is only guaranteed for $k \geq t - 3$.

As expected, larger values for k increase runtimes. For c17, k -instant diagnosis required up to 246 ms per instant for $k = 0$ and 775 ms for $k = 31$; for s27, up to 240 ms for $k = 0$ and 1106 ms for $k = 9$. Since k -temporal diagnosis treats each (c, t') as its own component, runtimes grow substantially: for one injected fault in c17, up to 8.7 s per instant were required for $k = 5$, compared to up to 136 s for $k = 12$. Because the circuits are Boolean, perfect pruning [14] retains complete information about instant variables $s^{t'}$ with $t' \in t \downarrow k$. With constant k over the trace, runtimes remain independent of trace length.

3.5 Extended Settings and Practical Considerations

Beyond the settings described here, our framework naturally captures additional scenarios, which we briefly outline and will study in future work:

Background Knowledge. In MBD, a component c known to be correct can be specified as background knowledge [27], reducing the diagnostic search space. In a LOLA DPI, this is achieved by extending α with $\wedge \neg c$.

Fault Models. If SD_{beh} (cf. Sec. 2.1) includes formulas $\text{AB}(c) \rightarrow fbeh_{c,1} \vee \dots \vee fbeh_{c,k}$ where $fbeh_{c,i}$ are different faulty behaviors, SD is a *strong fault model (SFM)* [40, Chap. 10]. This is naturally expressed in LOLA by extending α accordingly. E.g., if component T from Listing 2 is either stuck at 0 or outputs i_1 if abnormal, then $T \rightarrow (\text{out}_T = 0) \vee (\text{out}_T = i_1)$ is added to α . In this case, the different notion of a *kernel diagnosis* [10] (instead of minimal diagnosis) applies.

Sequential Diagnosis. Discrimination between multiple diagnoses can be done by automatically defining informative [28,34] monitors to prune the search space.

Assumptions about Faults. Diagnostic efficiency can be improved by constraining the state evolution of a component c . *Persistent faults*, which cannot recover automatically (e.g., a blown fuse), are specified by $c[-1|false] \rightarrow c$. *Time-invariant faults* are specified by $c[-1|false] \rightarrow c \wedge c \rightarrow c[-1|true]$.

Abstract Modeling. Besides precise value-based modeling, as discussed in this work, LOLA supports *abstract dependency-based models* (cf. [35]), both instead or combined with exact models. In this vein, a user can trade a higher diagnostic efficiency for a (potentially) higher number of spurious diagnoses [35].

Diagnostic Timing. Due to the inherent complexity of MBD [6], calculating diagnoses for every instant from can be practically hard or infeasible. Thus, a user can decide to diagnose only *periodically*, *on demand*, or *after fault detection*.

4 Conclusion

We introduced a unified framework integrating MBD and stream-based RV using LOLA. The approach enables online fault localization under possibly uncertain observations by way of symbolic reasoning. A proof-of-concept implementation confirms feasibility.

References

1. Alur, R.: Principles of cyber-physical systems. MIT press (2015)
2. Bauer, A., Leucker, M., Schallhart, C.: Model-based runtime analysis of distributed reactive systems. In: 17th Australian Software Engineering Conference (ASWEC 2006), 18-21 April 2006, Sydney, Australia. pp. 243–252. IEEE Computer Society (2006). <https://doi.org/10.1109/ASWEC.2006.36>, <https://doi.org/10.1109/ASWEC.2006.36>
3. Biere, A., Cimatti, A., Clarke, E.M., Strichman, O., Zhu, Y.: Bounded model checking. *Handbook of satisfiability* **185**(99), 457–481 (2009)
4. Brglez, F., Bryan, D., Kozminski, K.: Combinational profiles of sequential benchmark circuits. In: Proceedings of the IEEE International Symposium on Circuits and Systems. pp. 1929–1934 (1989). <https://doi.org/10.1109/ISCAS.1989.100747>
5. Brglez, F., Fujiwara, H.: A neutral netlist of 10 combinational benchmark circuits and a targeted translator in fortran. In: Proceedings of the IEEE International Symposium on Circuits and Systems (1985)
6. Bylander, T., Allemang, D., Tanner, M., Josephson, J.: The computational complexity of abduction. *Artificial Intelligence* **49**, 25–60 (1991). [https://doi.org/10.1016/0004-3702\(91\)90005-5](https://doi.org/10.1016/0004-3702(91)90005-5)
7. Cassandras, C.G., Lafortune, S.: Introduction to discrete event systems. Springer (2007)
8. D’Angelo, B., Sankaranarayanan, S., Sánchez, C., Robinson, W., Finkbeiner, B., Sipma, H.B., Mehrotra, S., Manna, Z.: LOLA: runtime monitoring of synchronous systems. In: Proc. of the 12th Int’l Symp. of Temporal Representation and Reasoning (TIME’05). pp. 166–174. IEEE Computer Society (2005). <https://doi.org/10.1109/TIME.2005.26>
9. Darwiche, A.: Decomposable negation normal form. *Journal of the ACM* **48**(4), 608–647 (Jul 2001). <https://doi.org/10.1145/502090.502091>
10. de Kleer, J., Mackworth, A.K., Reiter, R.: Characterizing diagnoses and systems. *Artificial Intelligence* **56**(2), 197–222 (1992). [https://doi.org/10.1016/0004-3702\(92\)90027-U](https://doi.org/10.1016/0004-3702(92)90027-U)
11. Feldman, A., Provan, G.M., van Gemund, A.J.C.: Computing Minimal Diagnoses by Greedy Stochastic Search. In: AAAI Conference on Artificial Intelligence (AAAI) (2008)
12. Friedrich, G., Lackinger, F.: Diagnosing Temporal Misbehavior. In: IJCAI. pp. 1116–1122 (1991)
13. Greiner, R., Smith, B.A., Wilkerson, R.W.: A correction to the algorithm in Reiter’s theory of diagnosis. *Artificial Intelligence* **41**(1), 79–88 (1989)
14. Hipler, R., Kallwies, H., Leucker, M., Montali, M., Sánchez, C., Winkler, S.: Symbolic runtime verification for monitoring under uncertainties and assumptions. *Information and Software Technology* **191**, 108004 (2026). <https://doi.org/10.1016/j.infsof.2025.108004>
15. Hipler, R., Kallwies, H., Leucker, M., Sánchez, C.: General anticipatory runtime verification. In: Proc. 36th CAV. LNCS, vol. 14682, pp. 133–155 (2024). https://doi.org/10.1007/978-3-031-65630-9_7
16. Ignatiev, A., Morgado, A., Weissenbacher, G., Marques-Silva, J.: Model-based diagnosis with multiple observations. In: International Joint Conference on Artificial Intelligence 2019. p. 1108–1115. Association for the Advancement of Artificial Intelligence (AAAI) (2019). <https://doi.org/10.24963/ijcai.2019/155>

17. Junker, U.: QUICKXPLAIN: Preferred Explanations and Relaxations for Over-Constrained Problems. In: McGuinness, D.L., Ferguson, G. (eds.) Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence. vol. 3, pp. 167–172. AAAI Press / The MIT Press (2004)
18. Kalech, M., Stern, R., Lazebnik, E.: Minimal cardinality diagnosis in problems with multiple observations. *Diagnostics* **11**(5), 780 (2021)
19. de Kleer, J., Williams, B.C.: Diagnosing multiple faults. *Artificial Intelligence* **32**(1), 97–130 (1987)
20. Köhl, M.A., Hermanns, H.: Model-based diagnosis of real-time systems: Robustness against varying latency, clock drift, and out-of-order observations. *ACM Trans. Embed. Comput. Syst.* **22**(4), 68:1–68:48 (2023). <https://doi.org/10.1145/3597209>
21. Leucker, M., Schallhart, C.: A brief account of runtime verification. *J. Logic Algebr. Progr.* **78**(5), 293–303 (2009)
22. Leucker, M., Schallhart, C.: A brief account of runtime verification. *The journal of logic and algebraic programming* **78**(5), 293–303 (2009)
23. Mealy, G.H.: A method for synthesizing sequential circuits. *The Bell System Technical Journal* **34**(5), 1045–1079 (1955). <https://doi.org/10.1002/j.1538-7305.1955.tb03788.x>
24. Metodi, A., Stern, R., Kalech, M., Codish, M.: A novel sat-based approach to model based diagnosis. *Journal of Artificial Intelligence Research* **51**, 377–411 (2014)
25. de Moura, L.M., Bjørner, N.: Z3: An efficient SMT solver. In: Proc. of the 14th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'08). LNCS, vol. 4963, pp. 337–340. Springer (2008). https://doi.org/10.1007/978-3-540-78800-3_24
26. Reiter, R.: A Theory of Diagnosis from First Principles. *Artificial Intelligence* **32**(1), 57–95 (1987)
27. Rodler, P.: Interactive Debugging of Knowledge Bases. Ph.D. thesis, Alpen-Adria Universität Klagenfurt (2015), <http://arxiv.org/pdf/1605.05950v1.pdf>
28. Rodler, P.: On Active Learning Strategies for Sequential Diagnosis. In: DX. pp. 264–283 (2017)
29. Rodler, P.: A formal proof and simple explanation of the QuickXplain algorithm. *Artificial Intelligence Review* **55**(8), 6185–6206 (Dec 2022). <https://doi.org/10.1007/s10462-022-10149-w>
30. Rodler, P.: Memory-limited model-based diagnosis. *Artificial Intelligence* **305**, 103681 (2022)
31. Rodler, P.: Random vs. Best-First: Impact of Sampling Strategies on Decision Making in Model-Based Diagnosis. In: AAAI Conference on Artificial Intelligence (AAAI) (2022)
32. Rodler, P.: DynamicHS: Streamlining Reiter's hitting-set tree for sequential diagnosis. *Information Sciences* **627**, 251–279 (2023)
33. Rodler, P.: How Should I Compute My Candidates? A Taxonomy and Classification of Diagnosis Computation Algorithms. In: ECAI 2023, pp. 1986–1993. IOS Press (2023)
34. Rodler, P.: Sequential model-based diagnosis by systematic search. *Artificial Intelligence* **323**, 103988 (2023)
35. Rodler, P., Hofer, B., Jannach, D., Nica, I., Wotawa, F.: Choosing abstraction levels for model-based software debugging: A theoretical and empirical analysis for spreadsheet programs. *Artificial Intelligence* p. 104399 (2025)
36. Rodler, P., Schekotihin, K.: Reducing Model-Based Diagnosis to Knowledge Base Debugging. In: DX. pp. 284–296 (2017)

37. Rodler, P., Teppan, E., Jannach, D.: Randomized problem-relaxation solving for over-constrained schedules. In: Int'l Conference on Principles of Knowledge Representation and Reasoning (KR). p. 696–701 (2021)
38. Shchekotykhin, K., Friedrich, G., Rodler, P., Fleiss, P.: Sequential diagnosis of high cardinality faults in knowledge-bases by direct diagnosis generation. In: ECAI'14. pp. 813–818 (2014)
39. Torasso, P., Torta, G.: Model-based diagnosis through OBDD compilation: A complexity analysis, p. 287–305. Springer (2006)
40. van Harmelen, F., Lifschitz, V., Porter, B.: Handbook of Knowledge Representation. Elsevier (2008)

A Proofs for Section 3

Proof of Lemma 1. Let Δ be a k -instant diagnosis for t . Furthermore, let $A := \{c^{t'} \mid c \in \Delta \wedge t' \in t \downarrow k\}$ and $B := \{\neg c^{t'} \mid c \in \text{COMPS} \setminus \Delta \wedge t' \in t \downarrow k\}$. By Def. 3, $A \cup B \cup [\varphi]_{\bar{\psi}}^{\leq t}$ is consistent. Since for every $t' \in t \downarrow k$, we have $\{c^{t'} \mid c \in \Delta\} \subseteq A$ and $\{\neg c^{t'} \mid c \in \text{COMPS} \setminus \Delta\} \subseteq B$, $\{c^{t'} \mid c \in \Delta\} \cup \{\neg c^{t'} \mid c \in \Delta\} \cup [\varphi]_{\bar{\psi}}^{\leq t'}$ must be consistent as well, and therefore Δ a 0-instant diagnosis for t' , again by Def. 3. \square

Proof of Theorem 1. (i) Since every $\Delta \in \mathbf{D}_{\text{all}}$ is by Lemma 1 a 0-instant diagnosis for all t' with $t' \in t \downarrow k$, which \mathbf{D}'_{all} contains all aggregations from, i.e., every $\Delta' \in \mathbf{D}'_{\text{all}}$ is also a 0-instant diagnosis for every t' , we have $\Delta \in \mathbf{D}'_{\text{all}}$.

(ii) The \subseteq direction follows from (i). Suppose for the other direction $\Delta \in \mathbf{D}'_{\text{all}}$. By definition of *aggr*, this implies Δ to be a 0-instant diagnosis for all t' , and, by Def. 3, $C_{t'} := \{c^{t'} \mid c \in \Delta\} \cup \{\neg c^{t'} \mid c \in \Delta\} \cup [\varphi]_{\bar{\psi}}^{\leq t'}$ to be consistent for those t' . Because φ is atemporal, each instant variable $x^{t'}$ does not have any relation to instant variables $y^{t''}$ with $t'' \neq t'$. This means that any set of constraints involving instant variables over multiple instants can be split into pairwise disjoint sets of constraints involving only instant variables of a single instant. Thus, the consistency of every $C_{t'}$ implies the consistency of $\bigcup_{t'=t-k}^t C_{t'} = \{c^{t'} \mid c \in \Delta \wedge t' \in t \downarrow k\} \cup \{\neg c^{t'} \mid c \in \text{COMPS} \setminus \Delta \wedge t' \in t \downarrow k\} \cup [\varphi]_{\bar{\psi}}^{\leq t}$ which means, by Definition 3, that $\Delta \in \mathbf{D}_{\text{all}}$.

(iii) Let $\Delta \in \mathbf{D}_{\text{min}}$. To contradict the proposition, there must not exist a $\Delta' \in \mathbf{D}'_{\text{min}}$ with $\Delta' \subseteq \Delta$, i.e., for all $\Delta' \in \mathbf{D}'_{\text{min}}$ it would hold that $\Delta \subset \Delta'$. However, since by (i) every k -instant diagnosis is also in *aggr*($\mathbf{D}_{t-k}, \dots, \mathbf{D}_t$), i.e., $\Delta \in \mathbf{D}'_{\text{all}}$, this implies the existence of a (non-minimal) aggregated diagnosis that is a subset of every minimal aggregated diagnosis. \square

Proof of Theorem 2. (i) By Definition 3, $\Delta \in \mathbf{D}_{\text{all}}$ implies that $C := \{c^{t'} \mid c \in \Delta \wedge t' \in t \downarrow k\} \cup \{\neg c^{t'} \mid c \in \text{COMPS} \setminus \Delta \wedge t' \in t \downarrow k\} \cup [\varphi]_{\bar{\psi}}^{\leq t}$ is consistent. Therefore, any subset $C' \subseteq C$ is also consistent. Let now $C' := \{c^{t'} \mid c \in \Delta \wedge t' \in t \downarrow k'\} \cup \{\neg c^{t'} \mid c \in \text{COMPS} \setminus \Delta \wedge t' \in t \downarrow k'\} \cup [\varphi]_{\bar{\psi}}^{\leq t}$ which, again by Def. 3, implies $\Delta \in \mathbf{D}'_{\text{all}}$.

(ii) Let $\Delta \in \mathbf{D}_{\text{min}}$. To contradict the proposition, there must not exist a $\Delta' \in \mathbf{D}'_{\text{min}}$ with $\Delta' \subseteq \Delta$, i.e., for all $\Delta' \in \mathbf{D}'_{\text{min}}$ it would hold that $\Delta \subset \Delta'$. However, since by (i) every k -instant diagnosis is also a k' -instant diagnosis, i.e.,

$\Delta \in \mathbf{D}'_{\text{all}}$, this implies the existence of a (non-minimal) k' -instant diagnosis that is a subset of every minimal k' -instant diagnosis. \square

Proof of Theorem 3. (i) (\subseteq) For each $\Delta \in \mathbf{D}_{\text{all}}$ there must exist a corresponding $\{(c, t') \mid c \in \Delta \wedge t' \in t \downarrow k\} \in \mathbf{D}'_{\text{all}}$ since by Definitions 3 and 4 both build the same constraints and must therefore both lead to consistency. (\supseteq) Let us assume there to be a $\Delta \in \mathbf{D}'_{\text{all}}$ with $\Delta' = \{c \mid (c, t') \in \Delta\} \notin \mathbf{D}_{\text{all}}$. Then, let $\Delta'' = \{(c, t') \mid c \in \Delta' \wedge t' \in t \downarrow k\} \supseteq \Delta$ and therefore $\Delta'' \in \mathbf{D}'_{\text{all}}$. Hence, Δ'' and Δ' build the same constraints again by Definitions 3 and 4, leading to $\Delta' \in \mathbf{D}_{\text{all}}$, and thus contradicting the assumption.

(ii) Given $\Delta \in \mathbf{D}_{\text{min}}$, $\Delta' = \{(c, t') \mid c \in \Delta \wedge t' \in t \downarrow k\}$ is by (i) a k -temporal diagnosis. While Δ' is not necessarily minimal, any minimal $\Delta'' \subseteq \Delta'$ ($\Delta'' \in \mathbf{D}'_{\text{min}}$) contains the same components as Δ' , only over possibly fewer timestamps. It is not possible that Δ'' contains fewer components than Δ' , i.e. $\{c \mid (c, t') \in \Delta''\} \subset \{c \mid (c, t') \in \Delta'\}$, because that would mean that $\{c \mid (c, t') \in \Delta''\}$ is, by (i), a k -instant diagnosis for t , leading to Δ not being minimal due to $\{c \mid (c, t') \in \Delta''\} \subset \Delta$, thus contradicting the assumption. Therefore, $\Delta = \{c \mid (c, t') \in \Delta''\} \in \text{tsFree}(\mathbf{D}'_{\text{min}})$. \square