



Synthesis of Temporal Causality

Bernd Finkbeiner[✉], Hadar Frenkel[✉], Niklas Metzger[✉],
and Julian Siber^(✉)[✉]

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
{finkbeiner, hadar.frenkel, niklas.metzger, julian.siber}@cispa.de

Abstract. We present an automata-based algorithm to synthesize ω -regular causes for ω -regular effects on executions of a reactive system, such as counterexamples uncovered by a model checker. Our theory is a generalization of *temporal causality*, which has recently been proposed as a framework for drawing causal relationships between trace properties on a given trace. So far, algorithms exist only for verifying a single causal relationship and, as an extension, cause synthesis through enumeration, which is complete only for a small fragment of effect properties. This work presents the first complete cause-synthesis algorithm for the class of ω -regular effects. We show that in this case, causes are guaranteed to be ω -regular themselves and can be computed as, e.g., nondeterministic Büchi automata. We demonstrate the practical feasibility of this algorithm with a prototype tool and evaluate its performance for cause synthesis and cause checking.

Keywords: Actual causality · Cause synthesis · Reactive systems · Temporal logic · Büchi automata

1 Introduction

Causality is a key ingredient for explaining model-checking results [5, 15, 38, 46] and a reasoning tool in several verification and synthesis algorithms [2, 36, 37]. These techniques have retrofitted causality definitions from philosophy [33, 40] and artificial intelligence [31], which were not designed for reactive systems with infinite dynamics and often fall short in such ad-hoc applications. For instance, popular approaches for explaining model-checking results highlight the counterexample trace at events that constitute causes [7, 18, 32]. Yet, marking a (possibly infinite) set of events does not clearly describe the temporal behavior manifested by them since, e.g., two events can be individually responsible for the effect or only together. Similarly, the occurrence of events in the loop part of a trace can be relevant, e.g., only once or infinitely often.

To address such reoccurring problems arising with causal reasoning in reactive systems, Coenen et al. have recently proposed *temporal causality* for drawing causal relationships between temporal properties on a given trace of a system [19]. Causal properties can then be described symbolically with logics or automata, which give a concise description of the possibly infinite causal behavior, and are, moreover, amenable to verification algorithms.

1.1 Temporal Causality

At its core, temporal causality uses counterfactual reasoning to infer a causal relationship: A property is a cause for some effect property on a given trace, where both properties hold, if on all closest traces that do not satisfy the cause, the effect is not satisfied either. Additionally, the cause property has to be semantically minimal. Hence, it is a form of *actual causation* [30], which describes the concrete causal behavior in the given, *actual* observation (the trace), and not all of the system behavior that may cause the effect (which loosely corresponds to the concept of *general causation*).¹

To illustrate, consider the system depicted in Fig. 1, where x and y are inputs and e is an output. We are interested in what input behavior causes the effect $\diamond e$ on the trace $\pi = (\{x, e\})^\omega$ – we skip the output label of the first position. Our first guess may be $y \vee \diamond x$, which characterizes all system traces that satisfy $\diamond e$. However, this is too general to describe the causal behavior on π . After all, the left disjunct y is not even satisfied by π . Let us see which condition fails. The counterfactual criterion holds: The closest system traces that do not satisfy $y \vee \diamond x$ also do not satisfy the effect, as these are exactly the

traces that go directly to the lower state labeled with the empty set and loop there infinitely. However, minimality is not satisfied, as the property $\diamond x$ implies $y \vee \diamond x$ (i.e., is semantically smaller) and also satisfies the counterfactual criterion: the closest trace that does not satisfy it is $(\{\})^\omega$. In particular, the existence of, e.g., trace $\{y, e\}(\{\})^\omega$ that also does not satisfy the cause $\diamond x$, but still satisfies the effect $\diamond e$, is irrelevant, as $(\{\})^\omega$ is closer to π than the trace $\{y, e\}(\{\})^\omega$. It is worth pointing out that we only measure distance over inputs. Picking a property that is *too* small fails the counterfactual criterion: If we picked $\square x$, which implies $\diamond x$, there would be, e.g., the closest trace $\{(\{x, e\})^\omega\}$ that still satisfies the effect.

In their original work [19], Coenen et al. showed that the requirements for a valid causal relationship can be encoded as a hyperproperty [17], such that checking whether a given ω -regular property is indeed the cause for a given ω -regular effect on a trace can be decided via model checking. This has recently been implemented in a sketch-based algorithm for enumerating causes [11], which is complete for effects containing \circ as the only temporal operator. That approach, of course, covers only a tiny fragment of the original theory. How to compute the cause for an arbitrary ω -regular effect has remained an open question.

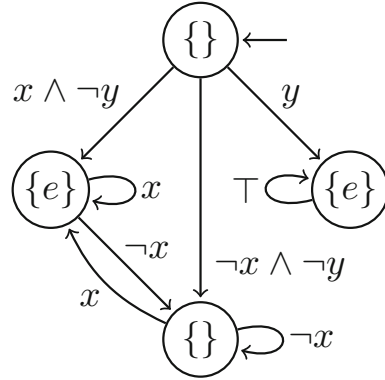


Fig. 1. Example system.

¹ Actual and general causality are also called token and type causality in the literature.

1.2 Contributions and Structure

As it turns out, the intricate balance between the counterfactual criterion and minimality of temporal causality gives rise to an intuitive order-theoretic characterization of causes: The complement of the cause is the upward closure of the negated effect property in the partial order defined by the similarity relation (measuring distance from the actual trace). We illustrate the intuition behind this characterization in Sect. 3.1, and formally develop it in Sect. 5.1.

The consequence of our characterization is that if we can compute the upward closure of the negated effect \bar{E} and the complement of the result, then we can compute the cause for E on π . We show that if E is an ω -regular property, π in a lasso shape, and the similarity relation is also defined by a (relational) ω -regular property, such an upward closure can be constructed as a nondeterministic Büchi automaton, which means that the cause (i.e., the complement of the automaton) again is an ω -regular property. This approach forms the core of our cause synthesis algorithm, which we describe in Sect. 5.

The complexity of our algorithm significantly scales in the size of the description of the similarity relation, which is problematic due to the complex and large similarity relations of previous work. Coenen et al. [19] observed that with the original counterfactual criterion, these similarity relations need to satisfy the assumption that there is a non-empty set of closest traces for any actual trace and candidate cause, otherwise the counterfactual condition can be vacuously true. We tie this restriction to the *limit assumption* first introduced by Lewis [41] and study similarity relations through this lens. Concrete similarity relations that have been proposed so far [11, 19] satisfy the limit assumption by adding additional criteria, but these increase the size of the formula describing the similarity relation significantly. In Sect. 4, we show that we can instead modify the counterfactual condition of the causality definition to allow similarity relations that do not satisfy the limit assumption, using Lewis’ semantics for counterfactuals [41], as extended to non-total similarity relations by Finkbeiner and Siber [23]. Crucially, this modification retains the original semantics of Coenen et al. for similarity relations that satisfy the limit assumption as long as the actual trace is deterministic. Hence, it generalizes our closure-based characterization and the corresponding algorithm to significantly simpler similarity relations. All proofs can be found in the full version of this paper [22].

In Sect. 6, we show through experiments with our prototype tool CORP that our modified counterfactual criterion leads to significantly faster computations in practice. We further compare our cause synthesis algorithm with the incomplete sketching approach of the tool CATS [11]. Last, we extend our approach to cause checking through cause synthesis with an additional equivalence check, which we compare with the checker implemented in CATS.

Contributions. In summary, we make the following contributions:

- We extend the theory of temporal causality to similarity relations that do not satisfy the limit assumption.

- We prove an order-theoretic characterization of causes as downward closed sets of the similarity relation.
- Based on this characterization, we develop the first complete method for ω -regular cause synthesis.
- We present and evaluate a prototype implementation of our approach.

2 Preliminaries

We start by recalling preliminaries regarding our system model. Then, we provide background on automata and logics for describing temporal properties.

Systems and Traces. We model *systems* as nondeterministic finite state machines $\mathcal{T} = (S, s_0, AP, \delta, l)$ where S is a finite set of *states*, $s_0 \in S$ is the *initial state*, $AP = I \cup O$ is the set of *atomic propositions* consisting of *inputs* I and *outputs* O , $\delta : S \times 2^I \rightarrow 2^S$ is the *transition function* determining a set of successor states for a given state and input, and $l : S \rightarrow 2^O$ is the *labeling function* mapping each state to a set of outputs. A *trace* of \mathcal{T} is an infinite sequence $\pi = \pi[0]\pi[1]\dots \in (2^{AP})^\omega$, with $\pi[i] = A \cup l(s_{i+1})$ for some $A \subseteq I$ and $s_{i+1} \in \delta(s_i, A)$ for all $i \geq 0$, i.e., we skip the label of the initial state in the first position. $\text{traces}(\mathcal{T})$ is the set of all traces of \mathcal{T} . For two subsets of atomic propositions $V, W \subseteq AP$, let $V|_W = V \cap W$, $\pi|_W = \pi_0|_W \pi_1|_W \dots$ and $\pi =_V \pi'$ iff $\pi|_V = \pi'|_V$ for traces π, π' . A trace π_0 is *deterministic* in \mathcal{T} iff for all $\pi_1 \in \text{traces}(\mathcal{T}) : \pi_0 =_I \pi_1 \rightarrow \pi_0 = \pi_1$. A trace π is *lasso-shaped*, if there exist $i, j = i + 1, k \in \mathbb{N}$ such that $\pi = \pi_0 \dots \pi_i \cdot (\pi_j \dots \pi_k)^\omega$, we then define $|\pi| = k - 1$.

Büchi Automata. A *nondeterministic Büchi automaton* (NBA) [13] is a tuple $\mathcal{A} = (Q, \Sigma, Q^0, F, \Delta)$, where Q denotes a finite set of *states*, Σ is a finite *alphabet*, $Q^0 \subseteq Q$ is a set of *initial states*, $F \subseteq Q$ is the set of *accepting states*, and $\Delta : Q \times \Sigma \rightarrow 2^Q$ is the *transition function* that maps a state and a letter to a set of possible successor states. The *size* of an NBA $|\mathcal{A}|$ is the number of its states $|Q|$. A *run* of \mathcal{A} on an infinite word $w = w_1 w_2 \dots \in \Sigma^\omega$ is an infinite sequence $r = q_0 q_1 \dots \in Q^\omega$ with $q_0 \in Q^0$ and $q_{i+1} \in \Delta(q_i, w_i)$ for all $i \in \mathbb{N}$. A run r of the NBA is *accepting* if there exist infinitely many $i \in \mathbb{N}$ such that $q_i \in F$. The *language* $\mathcal{L}(\mathcal{A})$ is the set of all words that have an accepting run. We say that some trace property $P \subseteq (2^A)^\omega$ is *ω -regular*, if there is an NBA \mathcal{A} such that $\mathcal{L}(\mathcal{A}) = P$. A trace π *satisfies* any $P \subseteq (2^A)^\omega$, denoted by $\pi \models P$, iff $\pi|_A \in P$.

Linear-Time Temporal Logic. We use *Linear-time Temporal Logic* (LTL) [44] to succinctly specify a fragment of ω -regular properties throughout the paper. LTL formulas are built using the following grammar, where $a \in AP$:

$$\varphi ::= a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \varphi \mathcal{U} \varphi \ .$$

The semantics of LTL are given by the following satisfaction relation, which recurses over the positions i of the trace π .

$$\begin{aligned}
 \pi, i \models a & \quad \text{iff } a \in \pi[i] \\
 \pi, i \models \neg\varphi & \quad \text{iff } \pi, i \not\models \varphi \\
 \pi, i \models \varphi \wedge \psi & \quad \text{iff } \pi, i \models \varphi \text{ and } \pi, i \models \psi \\
 \pi, i \models \bigcirc\varphi & \quad \text{iff } \pi, i + 1 \models \varphi \\
 \pi, i \models \varphi \mathcal{U} \psi & \quad \text{iff } \exists j \geq i \text{ such that } \pi, j \models \psi \text{ and } \forall i \leq k < j. \pi, k \models \varphi
 \end{aligned}$$

A trace π *satisfies* a formula φ , denoted by $\pi \models \varphi$ iff the formula holds at the first position: $\pi, 0 \models \varphi$. The *language* $\mathcal{L}(\varphi)$ is the set of all traces that satisfy a formula φ . We also consider the usual derived Boolean connectives: \vee , \rightarrow , \leftrightarrow ; and temporal operators: $\varphi \mathcal{R} \psi \equiv \neg(\neg\varphi \mathcal{U} \neg\psi)$, $\diamond\varphi \equiv \text{true} \mathcal{U} \varphi$, $\square\varphi \equiv \text{false} \mathcal{R} \varphi$.

Relational Properties. Relational properties, or, *hyperproperties* [17], allow us to relate multiple system executions, and reason about their interaction. Counterfactual reasoning often is a hyperproperty, and in particular, temporal causality as defined by Coenen et al. was formally shown to be a hyperproperty [19]. Many logics to express temporal hyperproperties have been suggested in recent years (e.g., [6, 8, 10, 28]), the most prominent one being HyperLTL [16]. In this paper, we do not use a hyperlogic to express temporal causality, but we use the related notion of *zipped traces* (e.g., [9]) for defining similarity relations. A *zipped trace* of three traces $\pi_{0,1,2}$ is defined as $\text{zip}(\pi_0, \pi_1, \pi_2)[i] = \{(a, t_k) \mid a \in \pi_k[i]\}$, i.e., we construct the zipped trace from disjoint unions of the positions of the three traces, where atomic propositions from the traces $\pi_{0,1,2}$ are distinguished through pairing them with the trace variables $t_{0,1,2}$.

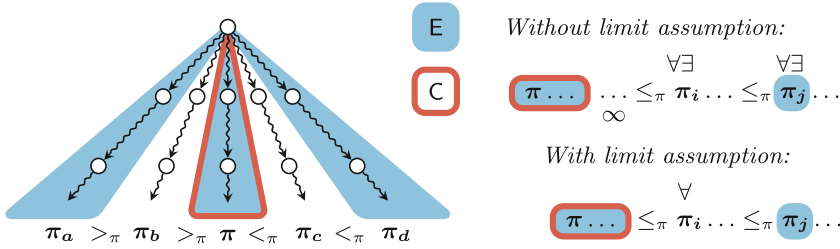
3 Overview: The Topology of Causality

Our main results on cause synthesis heavily rely on a characterization of causes as certain downward closed sets of system traces that are ordered by a similarity relation. We illustrate the main intuition behind this characterization in Sect. 3.1. Then, in Sect. 3.2, we outline how we extend this result to more general similarity relations than originally considered by Coenen et al. [19].

3.1 Actual Causes as Downward Closed Sets of Traces

Our central theorem states that the temporal cause for an effect E on some actual trace π is the largest subset of E that is downward closed² in the preordered set of system traces $(\text{traces}(\mathcal{T}), \leq_\pi)$, where \leq_π is a (comparative) similarity relation that orders traces based on their similarity to π . Figure 2a illustrates this abstractly. Arrows together with nodes represent system executions, whose

² $X \subseteq \text{traces}(\mathcal{T})$ is downward (upward) closed in $(\text{traces}(\mathcal{T}), \leq_\pi)$ if for all $\pi_x \in X$ and $\pi_t \in \text{traces}(\mathcal{T})$, $\pi_t \leq_\pi \pi_x$ ($\pi_x \leq_\pi \pi_t$) implies $\pi_t \in X$.



(a) The cause C as a subset of the effect E. (b) Possible situations at the limit of C.

Fig. 2. Two highlighted aspects of the cause C in the preordered set $(traces(\mathcal{T}), \leq_{\pi})$. Figure 2a illustrates that the cause is the largest downward-closed subset of the effect E. The quantifiers in Fig. 2b show which traces outside of the cause are required to avoid the effect in our formalization (*without limit assumption*) and in Coenen et al.’s definition [19] (*with limit assumption*). (Color figure online)

traces form $traces(\mathcal{T})$ and are ordered by the irreflexive reduction $<_{\pi}$ of the similarity relation. The set of system traces is, in general, infinite, such that there may be infinitely many other traces which are omitted from the illustration for sake of clarity. However, note that similarity relations must be designed such that all traces are further away from the actual trace π than itself, i.e., π is a minimum of \leq_{π} . The set of traces that satisfy the effect is depicted by the area that is colored in light blue. The actual trace π is an element of this set, as this is the trace on which the cause for a given effect is analyzed.

Coenen et al.’s temporal causality is counterfactual in nature, and now requires that the *closest* traces outside of the cause C, which in Fig. 2a is marked by the red border, do not satisfy the effect. In the illustration, this is reflected by π_b and π_c not satisfying the effect, i.e., not being in a light blue area. At the same time, Coenen et al. require the cause to be the smallest set that satisfies this, which means that only traces that satisfy the effect are included: Otherwise, the upward closure³ of traces that do not satisfy the effect could be removed. Hence, in Fig. 2a the area inside the red border is light blue.

In this paper, we show that the balance between these criteria defines causes that are the largest subsets of E that are downward closed in the preordered set $(traces(\mathcal{T}), \leq_{\pi})$. We also propose an algorithm that constructs these causes for effects that are ω -regular properties and traces that are in a lasso-shape. Our algorithm first constructs a nondeterministic Büchi automaton for the complement of the cause C. This complement is the upward closure of the negated effect \bar{E} , which means it includes all traces for which there exists an at-least-as close trace that does not satisfy the effect. Since \leq_{π} is reflexive, this naturally includes all traces in \bar{E} , such as π_b and π_c in Fig. 2a. It also includes all traces that are further away than a trace in \bar{E} , such as π_a and π_d .

³ The upward closure of a set X is the smallest upward closed set containing X.

In the end, these mechanisms make temporal causality a form of *actual* causality that describes a local generalization of the behavior that causes the effect on the actual trace. In the introductory example from Fig. 1 with the actual trace $\pi = \{x, e\}^\omega$, traces in, e.g., $\mathcal{L}(y \wedge \neg \diamond x)$ are all further away from π than the trace $\{\}^\omega$, which is in $\bar{E} = \mathcal{L}(\Box \neg e)$. Hence, $\mathcal{L}(y \wedge \neg \diamond x)$ is included in the upward closure of \bar{E} , and none of its elements is included in the cause.

3.2 Causality Without the Limit Assumption

With our approach based on set closure, we can solve a central issue of temporal causality: Since the preordered set $(\text{traces}(\mathcal{T}), \leq_\pi)$ is infinite, there only exist traces in \bar{C} that are the *closest* with respect to the actual trace π if (\bar{C}, \leq_π) is well-founded. If this is the case for all possible pairs of actual trace π and cause candidate C , we say the similarity relation satisfies the *limit assumption*, after Lewis [41], who formalized it for counterfactual modal logic. Since Coenen et al.’s definition [19] requires that all *closest* traces avoid the effect, it is restricted to similarity relations that satisfy this assumption. Their counterfactual condition is illustrated in the lower part of Fig. 2b. If the limit assumption holds, any descending chain $\pi_j \geq_\pi \pi_{j-1} \geq_\pi \dots$ stabilizes at some π_i , for which Coenen et al. require $\pi_i \in \bar{E}$.

If the limit assumption does not hold (upper part of Fig. 2b), there may be infinite chains $\pi_j \geq_\pi \pi_{j-1} \geq \dots$ for which a closest π_i does not exist. In these instances, Coenen et al.’s criterion would be vacuously true. This is particularly problematic as the canonical similarity relation \leq_π^{subset} does not satisfy the limit assumption. This metric orders two traces as $\pi_j \leq_\pi^{\text{subset}} \pi_k$ if the changes between π_j and π are a subset of the changes between π_k and π . This may lead, for example, to the infinite chain $\{\}^\omega \geq_\pi^{\text{subset}} \{x\} \{\}^\omega \geq_\pi^{\text{subset}} \dots$ in the preordered set $(\mathcal{L}(\Box \diamond \neg x), \leq_\pi^{\text{subset}})$, where $\pi = \{x\}^\omega$. Coenen et al. add additional constraints on top of \leq_π^{subset} to ensure that it satisfies the limit assumption. These, however, make cause checking more expensive, as observed by Beutner et al. [11], who therefore combine \leq_π^{subset} with a vacuity check. While this is computationally better, this check simply fails in instances as outlined above, and so certain causes cannot be checked by this method [11].

In this work, we solve this conundrum by modifying the definition of temporal causality to accommodate similarity relations that satisfy the limit assumption. We change the central counterfactual condition from a universal quantification over the closest traces in \bar{C} to an $\forall\exists$ -quantification over all traces $\pi_j \in \bar{C}$. For each such trace π_j , we require the existence of a closer trace $\pi_i \leq_\pi \pi_j$ that does not satisfy the effect. This is depicted in the upper part of Fig. 2b. Naturally, this quantification mirrors exactly the characterization of cause-complements via upward closed sets (cf. Sect. 5.1). On the theoretical side, we show that if the similarity relation satisfies the limit assumption and a minor assumption on nondeterminism is met, our definition is equivalent to Coenen et al.’s original definition (Sect. 4.3). On the practical side, we confirm experimentally that our approach leads to significant improvements through the accommodation of simpler similarity relations that do not satisfy the limit assumption (Sect. 6).

4 Generalized Temporal Causality

In this section, we generalize the definition of temporal causality to accommodate similarity relations that do not satisfy the limit assumption. We first recall similarity relations and formalize the limit assumption (Sect. 4.1). Then we present our updated definition of temporal causality (Sect. 4.2). Last, we prove that it retains the original semantics in the special case considered by Coenen et al. with a minor additional assumption on nondeterminism (Sect. 4.3).

4.1 Similarity Relations and the Limit Assumption

A *comparative similarity relation* $\leq_\pi \subseteq (2^I)^\omega \times (2^I)^\omega$ is a partial order that orders traces by their *comparative* distance from the given actual trace π , i.e., it gives no quantitative but a relative measurement of distance: $\pi_0 \leq_\pi \pi_1$ means π_0 is *at-least-as close* to π as π_1 . We measure distance over the set of inputs I , i.e., for two traces $\pi_{0,1} \in (2^{AP})^\omega$ we are only interested in $\pi_{0|I} \leq_\pi \pi_{1|I}$.

If I is clear from the context, we write $\pi_0 \leq_\pi \pi_1$. We require the actual trace to be closer to itself than any other trace, i.e., $\pi \leq_\pi \pi'$ for all $\pi' \in (2^{AP})^\omega$. The ternary relation \leq , where $(\pi_0, \pi_1, \pi_2) \in \leq$ iff $\pi_1 \leq_{\pi_0} \pi_2$, encodes the comparative similarity relations of all possible actual traces π_0 .

Example 1. To illustrate our formalism for similarity relations, consider the following *subset-based* similarity relation \leq^{subset} defined via the zipped trace $zip(\pi_0, \pi_1, \pi_2) \in (2^{AP \times \{t_0, t_1, t_2\}})^\omega$. To ease comprehension, for some $a \in AP$ we write a_{actual} for (a, t_0) , a_{close} for (a, t_1) , and a_{far} for (a, t_2) to explicitly identify, e.g., propositions on the actual trace, in a given formula. We then have $\pi_{close} \leq_{\pi_{actual}}^{subset} \pi_{close}$ iff

$$zip(\pi_{actual}, \pi_{close}, \pi_{far}) \models \bigwedge_{i \in I} ((i_{actual} \not\leftrightarrow i_{close}) \rightarrow (i_{actual} \not\leftrightarrow i_{far})) .$$

For the three traces $\pi_{actual}, \pi_{close}, \pi_{far}$ this requirement states that the changes between π_{actual} and π_{close} are a subset of the changes between π_{actual} and π_{far} , where we define the changes between two traces π_0, π_1 as $changes(\pi_0, \pi_1) = \{(a, i) \mid \pi_0[i] \neq_{\{a\}} \pi_1[i]\}$. For example, let $\pi = \{x\}(\{\})^\omega$, $\pi_0 = \{\}(\{\})^\omega$, $\pi_1 = \{\}\{y\}(\{\})^\omega$ and $I = \{x, y\}$. Then, $\pi_0 \leq_\pi \pi_1$, since $changes(\pi, \pi_0) = \{(x, 0)\} \subseteq \{(x, 0), (y, 1)\} = changes(\pi, \pi_1)$. The trace $\pi_2 = \{x\}(\{y\})^\omega$, however, is incomparable to π_0 and π_1 , as $changes(\pi, \pi_2) = \{(y, j) \mid j \geq 1\}$ is not in any subset relationship with the respective sets for π_0, π_1 .

The similarity relations considered in previous works [11, 19] are all fundamentally based on \leq^{subset} as defined in Example 1, with added conditions to avoid infinite chains of closer traces. This is directly tied to the *limit assumption* first studied by Lewis in his seminal work on counterfactual modal logic [41]. In our setting, this assumption can be formalized as follows.

Definition 1 (Limit Assumption). *A similarity relation $\leq \subseteq (2^I)^\omega \times (2^I)^\omega \times (2^I)^\omega$ satisfies the limit assumption, if for all traces $\pi \in (2^{I \cup O})^\omega$ and all possible causes $C \subseteq (2^I)^\omega$, we have that $(\bar{C}, <_\pi)$ is well-founded, i.e., there is no infinite descending chain $\pi_0 >_\pi \pi_1 >_\pi \dots$ with $\pi_i \in \bar{C}$.*

This requirement means that there always exist *closest* counterfactual traces that do not satisfy the cause no matter which actual trace we pick (except if all traces satisfy the cause). These closest traces would be ideal candidates for causal analysis, but unfortunately, they do not always exist, in particular not for the similarity relation \leq^{subset} , as stated in Proposition 1. Note that all proofs can be found in the full version of this paper [22].

Proposition 1. \leq^{subset} does not satisfy the limit assumption.

Since the original definition of Coenen et al. [19] quantifies universally over closest traces, it can be vacuously satisfied if the similarity relation does not satisfy the limit assumption. Previous works have therefore added additional constraints. For instance, Beutner et al. [11] propose \leq^{full} , which *additionally* to the constraints of \leq^{subset} (cf. Example 1) requires the following:

$$zip(\pi_{actual}, \pi_{close}, \pi_{far}) \models \bigwedge_{i \in I} (\Box \Diamond (i_{actual} \not\leftrightarrow i_{close}) \rightarrow \Box (i_{close} \leftrightarrow i_{far})) .$$

This encodes that whenever π_{close} differs from π_{actual} on some input at infinitely many locations, then π_{far} agrees with π_{close} on this input. Hence, on any chain in $<_\pi^{full}$, infinite changes on some $i \in I$ eventually get converted into finite ones, which ensures finiteness of the chain since there are only finitely many atomic propositions. We confirm that this results in \leq^{full} satisfying the limit assumption.

Proposition 2. \leq^{full} satisfies the limit assumption.

While satisfying the limit assumption is, in principle, useful, in the case of \leq^{full} this comes at a significant cost: Its logical description contains a large conjunction over the inputs, each containing an implication between temporal formulas. Hence, any algorithmic approach to cause synthesis (and checking) that uses \leq^{full} will scale poorly in the size of I . This motivates us to develop a modified definition of temporal causality that can directly work with the smaller, canonical similarity relation \leq^{subset} , while retaining most of the original semantics of Coenen et al. for similarity relations that satisfy the limit assumption, such as \leq^{full} .

4.2 A General Definition of Temporal Causality

We now develop our generalized definition of temporal causality for similarity relations that do not satisfy the limit assumption.

The idea behind our generalization stems from counterfactual modal logic as formalized by Lewis [41]. Lewis' semantics a priori only work for total similarity relations, making them unsuitable for our setting. However, they were recently extended to non-total similarity relations by Finkbeiner and Siber [23]. We apply these semantics to our concrete problem to obtain a well-defined notion of causality for similarity relations that do not satisfy the limit assumption. In Sect. 4.3, we show that our definition retains the original semantics proposed by Coenen et al. for similarity relations that satisfy the limit assumption.

Definition 2 (Temporal Causality). *Let \mathcal{T} be a system, $\pi \in \text{traces}(\mathcal{T})$ a trace, \leq_π a similarity relation, and $E \subseteq (2^{AP})^\omega$ an effect property. We say that $C \subseteq (2^I)^\omega$ is a cause of E on π in \mathcal{T} if the following conditions hold.*

- SAT:** For all $\pi_0 \in \text{traces}(\mathcal{T})$ such that $\pi_0 =_I \pi$ we have $\pi_0|_I \in C$ and $\pi_0 \in E$.
CF: For all $\pi_0 \in \bar{C}$ there is an at-least-as close trace $\pi_1 \in \bar{C}$, i.e., with $\pi_1 \leq_\pi \pi_0$, such that there is a $\pi_2 \in \text{traces}(\mathcal{T})$ with $\pi_1 =_I \pi_2$ and $\pi_2 \in \bar{E}$.
MIN: There is no $C' \subset C$ such that C' satisfies **SAT** and **CF**.

The main idea of the counterfactual criterion CF is that for every trace π_0 that does not satisfy the cause, there exists a closer trace π_2 that does not satisfy the cause *and* the effect. The additional quantification over π_1 is a technicality included because the cause $C \subseteq (2^I)^\omega$ consists of input sequences while $\pi_2 \in \text{traces}(\mathcal{T})$ is a full system trace. It also closely mirrors the structure of Coenen et al.'s PC2 criterion (cf. Definition 3) which it neatly generalizes to similarity relations that do not satisfy the limit assumption: If the assumption holds, then a π_2 is, in particular, required for the *closest* traces π_0 in \bar{C} , for which π_2 can only be instantiated by themselves. Hence, the closest traces are required to not satisfy the effect (we develop this comparison more formally in Sect. 4.3). If the limit assumption does not hold and there exists an infinite chain of ever-closer traces $\pi_0 \in \bar{C}$, the condition requires that for all these π_0 there is a closer π_2 that avoids the effect, even in infinity: No matter how far we descend on this chain, we are always guaranteed that we can descend further towards a closer counterfactual trace that does not satisfy the effect.

Example 2. To illustrate these conditions with a concrete example, consider the system from Fig. 1, the trace $\pi = \{x, e\}^\omega$, the effect $E = \mathcal{L}(\Box \Diamond e)$, and the cause $C = \mathcal{L}(\Box \Diamond x)$, with similarity relation \leq_{subset} . It is easy to see that SAT is satisfied, as the system is deterministic and $\pi|_I = \{x\}^\omega \in C$ and $\pi \in E$. There is, as discussed in Sect. 3.2, an infinite chain in $(\mathcal{L}(\Diamond \Box \neg x), \leq_\pi^{\text{subset}})$ and, hence, no closest trace. We require for all $\pi_0 \in \mathcal{L}(\Diamond \Box \neg x) = \bar{C}$ a $\pi_1 \in \bar{C}$ with $\pi_1 \in \mathcal{L}(\Diamond \Box \neg e) = \bar{E}$ and a $\pi_2 =_I \pi_1$ such that $\pi_2 \in \bar{E}$. In this case, we can pick π_1 as π_0 and π_2 as the corresponding system trace, hence CF is satisfied. To see that MIN is satisfied, consider any strict subset $C' \subset C$. Hence, there is some $\pi' \in C'$ such that $\pi' \models \Box \Diamond x$. Then, all system traces π_2 with $\pi_2 \leq_\pi \pi'$ satisfy $\pi_2 \models \Box \Diamond x$ by the definition of \leq_π^{subset} , and in this system this also means $\pi_2 \models \Box \Diamond e$. Hence, C satisfies MIN because no strict subset satisfies CF.

Remark 1. Note that Definition 2 is not restricted to similarity relations that can be expressed via zipped traces and LTL formulas as used in the previous examples, but instead applies to any comparative similarity relation as defined at the start of this section.

4.3 Proving Generalization

This section is dedicated to proving that our generalization (Definition 2) is conservative, i.e., agrees with Coenen et al.’s original definition whenever the underlying similarity relation satisfies the limit assumption and the actual trace is deterministic. First, we recall Coenen et al.’s definition.

Definition 3 (Coenen et al. [19]). *Let \mathcal{T} be a system, $\pi \in \text{traces}(\mathcal{T})$ a trace, \leq_π a similarity relation, and $E \subseteq (2^O)^\omega$ an effect property. $C \subseteq (2^I)^\omega$ is a cause of E on π in \mathcal{T} if the following three conditions hold.*

PC1: $\pi|_I \in C$ and $\pi \in E$.

PC2: *For all closest counterfactual traces $\pi_0 \in \bar{C}$, i.e., traces for which there are no closer traces $\pi_1 \in \bar{C}$ with $\pi_1 <_\pi \pi_0$, there exists a $\pi_2 \in \text{traces}(\mathcal{T})$ such that $\pi_0 =_I \pi_2$ and $\pi_2 \in \bar{E}$.*

PC3: *There is no $C' \subset C$ such that C' satisfies **PC1** and **PC2**.*

Unlike in our updated definition, PC1 only works if the actual trace π is deterministic. If the π is nondeterministic, the effect can be avoided with no modifications at all to π (which is minimal), hence the cause should be empty. PC1 does not reflect this and allows to build a cause that includes $\pi|_I$ (and possibly more), wrongfully implying that a modification of the sequence is required to avoid the effect. PC2 may be vacuously satisfied if the similarity relation does not satisfy the limit assumption, as outlined in Sect. 3.2.

Remark 2. Note that Coenen et al. consider traces $\pi \in \text{traces}(\mathcal{C}_\pi^T)$ of the counterfactual automaton \mathcal{C}_π^T for PC2. This automaton models *contingencies*, which allow to partially reset outputs back to as they were on the actual trace π , and to change the system state accordingly. For PC2 in Definition 3, this means that the closest counterfactual traces π_2 do not have to avoid the effect themselves, but together with some contingency. This mechanism, inspired by Halpern’s modified version of actual causality [29], was extended by Coenen et al. [18, 19] to lasso-shaped traces and finite state machines to sometimes obtain more accurate causes. However, to guarantee meaningful results, the original system has to have unique output labels. Beutner et al.’s implementation [11] therefore allows to toggle the usage of contingencies. Similarly, our generalization works both with contingencies and without. For the latter case, one simply supplants \mathcal{T} with \mathcal{C}_π^T in both definitions. Our cause synthesis algorithm can also handle contingencies, and our implementation allows to toggle them as a feature. Our theoretical contribution is independent of this detail.

We now proceed to show the equivalence between our definition (Definition 2) and Coenen et al.'s definition (Definition 3) in case the limit assumption is fulfilled and the actual trace is deterministic. We start with proving the equivalence of the counterfactual conditions CF and PC2, which holds regardless of nondeterminism on the actual trace.

Lemma 1. *Let \mathcal{T} be a system, $\pi \in \text{traces}(\mathcal{T})$ a trace, $\mathsf{C} \subseteq (2^I)^\omega$ a cause property, and $\mathsf{E} \subseteq (2^{AP})^\omega$ an effect property. Let \leq be a similarity relation that satisfies the limit assumption. Then we have that PC2 is satisfied iff CF is satisfied.*

With Lemma 1 at hand, we only need to address the differences between PC1 and SAT. It is easy to see that their equivalence fails when behavior on the actual trace π is nondeterministic, i.e., when there is another trace that is input-equivalent to π but does not satisfy the effect. In such a case, PC1 is satisfied but SAT is not. Hence, our definition is equivalent to Coenen et al.'s definition only in deterministic systems, as we deliberately diverge in the case of nondeterminism on the actual trace. Notably, Lemma 1 holds for both deterministic and nondeterministic systems, and determinism is only relevant on the actual trace. The restriction to output-only effects $\mathsf{E} \subseteq (2^O)^\omega$ is inherited from Coenen et al.'s definition, but technically not necessary.

Theorem 1. *Let \leq be a similarity relation that satisfies the limit assumption. Then $\mathsf{C} \subseteq (2^I)^\omega$ is a cause for $\mathsf{E} \subseteq (2^O)^\omega$ on a trace π that is deterministic in \mathcal{T} according to our definition (Definition 2) if and only if it is a cause according to Coenen et al.'s definition (Definition 3).*

5 Cause Synthesis

In this section, we develop our algorithm for synthesizing causes. In Sect. 5.1 we formalize the characterization of a cause as the complement of the upper closure of the negated effect, which we have discussed intuitively in Sect. 3.1. In Sect. 5.2 we provide an algorithm for cause synthesis in the ω -regular setting, when the effect is given as a nondeterministic Büchi automaton and the actual trace is in a lasso shape.

5.1 Proving Our Characterization

For this section, we fix a system \mathcal{T} , an actual trace $\pi \in \text{traces}(\mathcal{T})$, a similarity relation \leq , and an effect E . We now show that, if it exists, the cause for E on π is the complement of the upward closure of $\bar{\mathsf{E}}$ in $(\text{traces}(\mathcal{T}), \leq_\pi)$. Formally, we construct a set D that is a cause for E on π via its complement:

$$\begin{aligned} \bar{\mathsf{D}} &= \{ \rho \in (2^I)^\omega \mid \exists \sigma \in \text{traces}(\mathcal{T}). \sigma \leq_\pi \rho \wedge \sigma \in \bar{\mathsf{E}} \} , \text{ hence} \\ \mathsf{D} &= \{ \rho \in (2^I)^\omega \mid \forall \sigma \in \text{traces}(\mathcal{T}). \sigma \leq_\pi \rho \rightarrow \sigma \in \mathsf{E} \} . \end{aligned}$$

The set D directly corresponds to the (unique) cause if there exists one, and is empty if there is none. We establish this in a series of lemmas.

Lemma 2. *If the set D is non-empty, it is a cause for E on π in \mathcal{T} .*

Lemma 2 shows that D satisfies Definition 2 assuming it is non-empty. The assumption is only required for SAT, as this criterion requires that π and all input-equivalent traces are in the cause. CF follows from the definition of D , and for MIN we can show that any strict subset of D does not satisfy CF.

Lemma 3. *If the set D is empty, there exists no cause that satisfies SAT.*

Lemma 3 serves two purposes. First, it helps us argue for the completeness of our construction. Second, it shows that the only reason why there may be no cause is due to a nondeterministic actual trace. To fully argue completeness, we show that causes are unique, and hence D is the only relevant cause in all cases.

Lemma 4. *Causes are (semantically) unique: There can be no two sets $C \neq C'$ that are both causes for some effect property E on a trace π in some system \mathcal{T} .*

Remark 3. This does *not* mean that there can only exist a single causal event, such as “ a at position 0” or “ b at position 1”, in a given scenario. Instead, Lemma 4 states that the semantics of the symbolic description of the causal behavior in a given scenario is unique. It is precisely the idea of temporal causality to encompass multiple single events in a single symbolic description, e.g., through a conjunction such as $a \wedge \circ b$.

5.2 Cause-Synthesis Algorithm for ω -Regular Effects

In Sect. 5.1, we have established a direct characterization of causes as downward closed sets, independent of any concrete descriptions of cause, effect, and trace. In this section, we develop an automata-based algorithm for synthesizing causes of ω -regular effects given, e.g., by a nondeterministic Büchi automaton (NBA), on lasso-shaped traces. We assume that the relation $\leq \subseteq (2^I)^\omega \times (2^I)^\omega \times (2^I)^\omega$ is definable by a relational ω -regular property $P_\leq \subseteq (2^{I \times \{t_0, t_1, t_2\}})^\omega$, such that $(\pi_0, \pi_1, \pi_2) \in \leq$ iff the zipped trace $\text{zip}(\pi_0, \pi_1, \pi_2)$ satisfies P_\leq . Note that this applies to all concrete similarity relations introduced in Sect. 4. We show that under these assumptions, the set D from Sect. 5.1 can be constructed as an NBA. First, we construct an NBA for \bar{D} and subsequently complement it. This is necessary because we start out from an NBA representation for the effect, and assume the similarity relation to be given by an NBA as well. Since the NBAs acceptance condition is existential, we need the additional complementations to express the universal quantification over the closer traces σ appearing in the definition of D .

The main technical difficulty that remains is to ensure that the conditions on the three traces π_{actual} , π_{close} and π_{far} , as they appear in the alphabet of a similarity relation, are applied consistently, and that the quantification over σ in D , which corresponds to π_{close} , is resolved at the correct step, as the automaton should range over the inputs I and not, e.g., over $I \times \{t_0, t_1, t_2\}$ as used by the similarity relation.

Similarity Relation. Our starting point is the NBA for the similarity relation defined by the ω -regular property P_{\leq} : $\mathcal{A}_{\leq}^I = (Q_{\leq}, 2^{I \times \{t_0, t_1, t_2\}}, Q_{\leq}^0, F_{\leq}, \Delta_{\leq}^I)$. The automaton \mathcal{A}_{\leq}^I only reasons about inputs and uses tuples with the trace variables t_0, t_1 and t_2 to encode whether the input appears on the actual, closer or farther trace, respectively. We lift the automaton to the full set of atomic propositions as the automaton $\mathcal{A}_{\leq} = (Q_{\leq}, 2^{(I \times \{t_0, t_1, t_2\}) \cup (O \times \{t_0, t_1\})}, Q_{\leq}^0, F_{\leq}, \Delta_{\leq})$. The transition relation is defined as follows, for a letter w : $q_2 \in \Delta_{\leq}(q_1, w)$ iff $q_2 \in \Delta_{\leq}^I(q_1, w \setminus (O \times \{t_0, t_1\}))$. Hence, \mathcal{A}_{\leq} specifies the same relation between the inputs of the three traces as \mathcal{A}_{\leq}^I , but allows arbitrary output behavior. Its alphabet does not contain outputs for π_2 , as these traces eventually form the elements of the cause, which only ranges over the inputs.

Effect. Next, we modify the NBA $\mathcal{A}_{\mathbf{E}}^* = (Q_{\mathbf{E}}, 2^{AP}, q_{\mathbf{E}}, F_{\mathbf{E}}, \Delta_{\mathbf{E}}^*)$ for the ω -regular effect \mathbf{E} such that it refers to the closer trace t_1 and ranges over the same alphabet as \mathcal{A}_{\leq} . We obtain $\mathcal{A}_{\mathbf{E}} = (Q_{\mathbf{E}}, 2^{(I \times \{t_0, t_1, t_2\}) \cup (O \times \{t_0, t_1\})}, q_{\mathbf{E}}, F_{\mathbf{E}}, \Delta_{\mathbf{E}})$ with:

$$\begin{aligned} q_2 \in \Delta_{\mathbf{E}}(q_1, (w \times \{t_1\}) \cup X \cup Y) \text{ iff} \\ q_2 \in \Delta_{\mathbf{E}}^*(q_1, w) \wedge X \subseteq (AP \times \{t_0\}) \wedge Y \subseteq (I \times \{t_2\}) . \end{aligned}$$

Hence, $\mathcal{A}_{\mathbf{E}}$ restricts π_1 to be in \mathbf{E} by restricting it to the transition relation of $\mathcal{A}_{\mathbf{E}}^*$, while allowing an arbitrary trace π_0 and arbitrary input sequence in π_2 .

Intersection. For the conjunction that defines the set $\bar{\mathbf{D}}$, we intersect \mathcal{A}_{\leq} with the complement of $\mathcal{A}_{\mathbf{E}}$ to obtain $\mathcal{A}_{\cap} = (Q_{\cap}, 2^{(I \times \{t_0, t_1, t_2\}) \cup (O \times \{t_0, t_1\})}, Q_{\cap}^0, F_{\cap}, \Delta_{\cap})$ such that: $\mathcal{A}_{\cap} = \mathcal{A}_{\leq} \cap \bar{\mathcal{A}}_{\mathbf{E}}$.

System Product. As the next step, we construct the product of the automaton \mathcal{A}_{\cap} with the system $\mathcal{T} = (S, s_0, AP, \delta, l)$, ensuring that the atomic propositions t_1 are picked from a valid system trace. When building the product, we project away explicit atomic propositions paired with t_1 , as the traces of the desired set \mathbf{D} are only the traces paired with t_2 . The resulting automaton is $\mathcal{A}_{\times} = (S \times Q_{\cap}, 2^{(I \times \{t_0, t_2\}) \cup (O \times \{t_0\})}, \{s_0\} \times Q_{\cap}^0, S \times F_{\cap}, \Delta_{\times})$, where

$$\begin{aligned} \Delta_{\times}((s_i, q_i), w) = \{ (s_{i+1}, q_{i+1}) \mid \exists A \subseteq I. s_{i+1} \in \Delta_{\cap}(s_i, A) \\ \wedge q_{i+1} \in \Delta_{\cap}(q_i, (l(s_{i+1}) \cup A) \times \{t_1\}) \} . \end{aligned}$$

Cause Automaton. To obtain the final result, we first complement the automaton from the previous step to obtain $\bar{\mathcal{A}}_{\times} = (Q_{\times}, 2^{(I \times \{t_0, t_2\}) \cup (O \times \{t_0\})}, Q_{\times}^0, F_{\times}, \Delta_{\times})$, and then build the product with the trace. At the same step we project away atomic propositions paired with t_0 , and remove the trace variable t_2 to obtain the alphabet 2^I for the cause. For the lasso-shaped trace $\pi = \pi_0 \dots \pi_{j-1} \cdot (\pi_j \dots \pi_k)^{\omega}$ we define the set of positions as $\Pi = \{\pi_0, \dots, \pi_k\}$ and a successor function $\text{succ} : \Pi \mapsto \Pi$ as $\text{succ}(\pi_r) = \pi_{r+1}$ for $r < k$, and $\text{succ}(\pi_k) = \pi_j$. The cause automaton is then $\mathcal{A}_{\mathbf{D}} = (\Pi \times Q_{\times}, 2^I, \{\pi_0\} \times Q_{\times}^0, \Pi \times F_{\times}, \Delta_{\mathbf{D}})$, where

$$\Delta_{\mathbf{D}}((\pi_i, q_i), w) = \{ (\text{succ}(\pi_i), q_{i+1}) \mid q_{i+1} \in \Delta_{\times}(q_i, (\pi_i \times \{t_0\}) \cup (w \times \{t_2\})) \} .$$

From the lemmas established in Sect. 5.1, we conclude that there is a cause iff \mathcal{A}_D is non-empty, and then the cause is uniquely determined by its language.

Corollary 1. *The language of \mathcal{A}_D is empty iff there is no cause C for E on π in \mathcal{T} , and if $\mathcal{L}(\mathcal{A}_D)$ is non-empty, then it is the unique cause for E on π in \mathcal{T} .*

We can also state an upper bound on the size of \mathcal{A}_D , which is dominated by the potentially exponential growth from NBA complementation [47].

Proposition 3. *If the effect E and the similarity relation \leq are given as NBAs \mathcal{A}_E and \mathcal{A}_{\leq} , respectively, then the size of \mathcal{A}_D is in $|\pi| \cdot 2^{(2^{O(|\mathcal{A}_E|)} \cdot |\mathcal{A}_{\leq}| \cdot |\mathcal{T}|)}$.*

Note that the doubly-exponential upper bound in the description of E persists independent of whether it is given as an NBA or LTL formula. In the latter case, we simply translate the negated formula, which again leads to an exponential blow-up. In theory, the description does make a difference for \leq : If it is represented as a formula, we first need to translate it with a potentially exponential increase in size, hence it would move up one exponent in the bound. In practice, the canonical similarity relation \leq^{subset} can always be represented by a 1-state NBA, such that its contribution to the bound is less relevant.

While the stated upper bound may seem daunting, it mirrors the (tight) bounds of related problems, such as LTL synthesis [45]. In the following section, we show that, not only can our approach solve many cause-synthesis problems in practice, it also significantly improves upon previous methods for cause checking.

6 Implementation and Evaluation

In this section, we evaluate a prototype tool implementing our cause-synthesis approach, called CORP - *Causes for Omega-Regular Properties*.⁴ Our prototype is written in Python and uses Spot [21] for automata operations and manipulation. The prototype allows for both cause synthesis and cause checking, where in the latter case the correct cause is first synthesized and then checked for equivalence against the cause candidate. This allows for a direct comparison with the cause checking tool CATS [11] in Sect. 6.2. Before, we report on our experiments on cause synthesis, where we compare our method with the incomplete, sketch-based approach of CATS. All experiments were carried out on a machine equipped with a 2.8 GHz Intel Xeon processor and 64 GB of memory, running Ubuntu 22.04.

6.1 Cause Synthesis

We conducted three different experiments that highlight how the similarity relations, effect size and system size contribute to the performance of our algorithm.

⁴ Our prototype is on GitHub: <https://github.com/reactive-systems/corp>. Our full evaluation can be reproduced with the artifact on Zenodo: <https://doi.org/10.5281/zenodo.10946309>.

Table 1. Cause synthesis for arbiters. $|\mathcal{T}|$ is the number of system states. In all instances, π is the (unique) trace where all n clients send a request at every position, which has length $|\pi| = n$. φ_E is the effect. We report the time taken to synthesize the causal NBA with the metrics \leq^{full} and \leq^{subset} in seconds and the respective NBA sizes $|\mathcal{A}_C^{full}|$ and $|\mathcal{A}_C^{subset}|$, and provide an LTL description φ_C of the NBA language (guessed manually). TO denotes the timeout of 60 seconds.

Instance	$ \mathcal{T} $	φ_E	$t(\leq^{full})$	$t(\leq^{subset})$	$ \mathcal{A}_C^{full} $	$ \mathcal{A}_C^{subset} $	φ_C
SPURIOUS 1	1	$\diamond g_0$	0.11	0.11	1	1	<i>true</i>
SPURIOUS 2	2	$\diamond g_0$	0.11	0.11	1	1	<i>true</i>
SPURIOUS 3	3	$\diamond g_0$	0.21	0.11	1	1	<i>true</i>
SPURIOUS 4	4	$\diamond g_0$	TO	0.11	-	1	<i>true</i>
UNFAIR 2	2	$\square \neg g_0$	0.11	0.11	1	1	$\square r_{prio}$
UNFAIR 3	4	$\square \neg g_0$	0.16	0.11	1	1	$\square r_{prio}$
UNFAIR 4	6	$\square \neg g_0$	TO	0.11	-	1	$\square r_{prio}$
FULL 1	1	$\diamond g_0$	0.11	0.11	2	2	$\diamond r_0$
		$\square \diamond g_0$	0.11	0.11	2	2	$\square \diamond r_0$
FULL 2	4	$\diamond g_0$	0.11	0.11	2	2	$\diamond r_0$
		$\square \diamond g_0$	0.11	0.11	12	4	$\square \diamond r_0$
FULL 3	11	$\diamond g_0$	0.16	0.11	2	2	$\diamond r_0$
		$\square \diamond g_0$	2.04	0.16	215	24	$\square \diamond r_0$
FULL 4	46	$\diamond g_0$	TO	0.11	-	2	$\diamond r_0$
		$\square \diamond g_0$	TO	33.22	-	214	$\square \diamond r_0$

Arbiters. We computed causes on traces of resource arbiters to compare the performance of our algorithm under different similarity relations, whose logical description scales in the number of system inputs. An arbiter instance is parameterized by a number of clients n , each with its own input. This let us easily scale the size of the similarity relation’s description. For some number n of clients (indexed by k) that request access to a shared resource with a request r_k , an arbiter grants mutually exclusive access to the resource with a grant g_k . We considered different arbiter strategies, and for each we synthesize causes as NBAs \mathcal{A}_C^{full} and \mathcal{A}_C^{subset} with the similarity relations \leq^{full} and \leq^{subset} , respectively. The results of these instances are depicted in Table 1. Spurious arbiters simply give out grants to all clients in a round-robin manner, regardless of previous requests. Unfair arbiters prioritize one client with request r_{prio} over the others, while full arbiters are fully functional arbiters that only give out grants that were requested beforehand. In all instances, we computed causes on the (unique) trace π where all clients send requests continuously, i.e., $\pi|_I = \{r_0, \dots, r_n\}^\omega$. Consequently, on this trace both the spurious and the full arbiter send grants to all clients, while the unfair arbiter only gives grants to the prioritized client. These varying strategies are reflected in the synthesized cause-effect pairs. In the spurious arbiters, the language of the synthesized cause NBA for the effect $\diamond g_0$ is *true*,

which reflects that the effect appears on *all* system traces. In the unfair arbiters, the cause for no grant being given to client 0 is that the prioritized arbiter sends requests permanently, i.e., the causal NBA has the language $\square r_{prio}$. In the full arbiters, $\diamond g_0$ is caused, as expected, by $\diamond r_0$ and $\square \diamond g_0$ is caused by $\square \diamond r_0$. From a performance standpoint, the arbiter instances show us that accommodating the canonical similarity relation \leq^{subset} , as we did through our generalization of temporal causality in Sect. 4, leads to significant improvements in practice: In all instances, synthesizing causes with \leq^{subset} was faster than with \leq^{full} , and the resulting causal NBAs were smaller as well. This is mostly because of the number of inputs involved: The other parameters stay comparably small when going from the spurious 1-arbiter to the spurious 4-arbiter, but the latter times out when using \leq^{full} . When the systems get larger and the effects more complex, e.g., in the instance of the full 4-arbiter with the effect $\square \diamond g_0$, the automata produced can become bigger even with \leq^{subset} . However, the language of the produced automata has a small representation, i.e., $\square \diamond r_0$, such that we see potential for improvement through automata minimization techniques.

Neural Synthesis. For more diverse effects, we considered mispredicted circuits from a neural synthesis model [48]. Given some specification (in this case, generated by Spot’s `randltl`) the neural model predicts an implementation as an AIGER [12] circuit, which is in the end model-checked against the specification. Since neural synthesis is not sound, this check fails occasionally and returns a counterexample, which may be used for further repair [20]. We used our tool CORP to compute the cause for the violation of the specification on such a counterexample. In Fig. 3 we report the time of computing causes with respect to size of the syntax tree of the effect formula, and the system size. The timeout was set to 100s. The size of the points in the scatter plot corresponds to the length of the counterexample and the color to the system size. From the plot we can deduce that a large effect does not mean a long runtime of our tool per se. However, a combination of large effects, bigger systems, and longer counterexamples usually means that the tool takes longer. The sizes of the synthesized causes are diverse and range from 2 to 60 states.

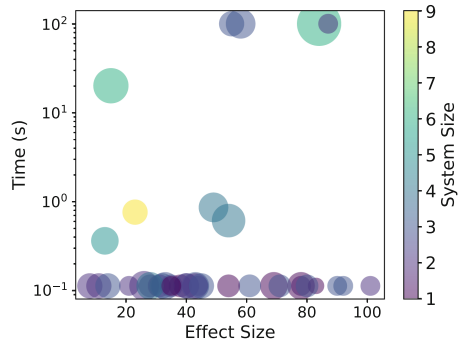


Fig. 3. Computing causes for neural synthesis mispredictions with CORP. Size of a point represents the length of the counterexample (between 2 and 16).

Example 3. We discuss an illustrative example of cause synthesis with a small benchmark from the neural synthesis dataset. All relevant inputs and outputs of

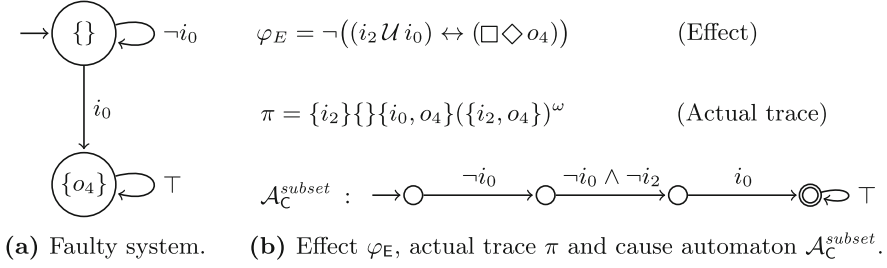


Fig. 4. A system predicted wrongly by a neural synthesis model (Fig. 4a) for the specification $\neg\varphi_E$, i.e., the negation of the effect. The effect is shown together with the actual trace π , i.e., a counterexample obtained from model checking, and the computed cause automaton \mathcal{A}_C^{subset} in Fig. 4b.

our cause synthesis algorithm are depicted in Fig. 4. First, we have the system (cf. Fig. 4a), which is a wrongly predicted circuit of the neural synthesis model. This model tried to come up with a solution for the specification $(i_2 \mathcal{U} i_0) \leftrightarrow (\Box \Diamond o_4)$, i.e., o_4 appears infinitely often if and only if input i_2 is enabled until input i_0 is enabled. The predicted system does not satisfy this specification, because there are cases where $\Box \Diamond o_4$ holds without the inputs meeting the required condition. Hence, model checking the specification returns a counterexample π that violates the formula, which means the negated specification can be seen as an effect φ_E that is present on the counterexample π (cf. Fig. 4b). Our algorithm then computes the cause for this effect, i.e., for the violation of the specification, on the counterexample π , as a nondeterministic Büchi automaton. The computed automaton \mathcal{A}_C^{subset} is depicted at the bottom of Fig. 4b. It is language-equivalent to the LTL formula $\neg i_0 \wedge \bigcirc (\neg i_0 \wedge \neg i_2 \wedge \bigcirc i_0)$, which basically states that the effect is caused by a conjunction of four inputs spread out over the first three steps. Indeed, it is easy to see that modifying any of these four inputs results in a trace that satisfies the specification: For instance, setting i_0 at the first position results in the trace that immediately enters the state labeled with o_4 and loops there forever such that the left part of the equivalence is satisfied, while removing i_0 from the third position results in looping in the initial state such that the right part of the equivalence is not satisfied anymore.

Comparison with Cause Sketching. CATS, the tool of Beutner et al. [11], allows to enumerate non-temporal formulas in holes of a provided cause sketch until a cause is found. If the effect contains \bigcirc as the only temporal operator and a cause exists, there is a sketch that is guaranteed to encompass the cause. This provides us with a baseline with which we can compare our cause-synthesis algorithm. We constructed random benchmarks that fall into CATS’ complete fragment using Spot’s `randaut` function to generate systems with 10 up to 1000 states, obtaining traces of length 2 and then inserting a new atomic proposition e at the last position of the trace and in the system. The effect then is defined as the occurrence of e at exactly this position. We chose such small traces and effects

because CATS timed out already on slightly larger instances. We conducted additional experiments using just our tool CORP with traces (and effects) of size 10. Figure 5a shows the time taken by CATS and CORP to synthesize causes. The influence of the system size on the runtime of CORP in this setting is negligible, which we believe is due to the efficient automata operations performed by Spot. The hyperproperty encoding of CATS does not seem as amenable to similar optimizations.

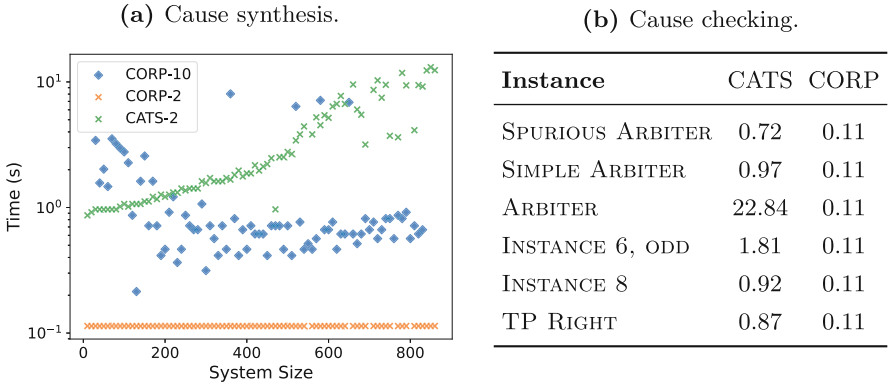


Fig. 5. Direct comparisons between our tool CORP and the tool CATS [11]. Figure 5a shows the time CATS needs to synthesize a cause in its complete fragment with trace and effect of size 2, and the time taken by CORP for sizes 2 and 10. Figure 5b shows the time taken to check single causal relationships. These problems are taken from Beutner et al. [11] (where “Instances” are “Examples”).

6.2 Cause Checking

It is straightforward to use our cause synthesis algorithm to also check causes through an equivalence check between the synthesized causal NBA and the candidate formula (or automaton). This allows a direct performance comparison with the cause checking tool CATS of Beutner et al. [11], which we conducted on the publicly available benchmarks of their paper. In these cause-checking benchmarks, a cause candidate is given in addition to the system, actual trace and effect. The time CATS and our tool CORP took in each instance to check whether the given candidate is a cause is depicted in Fig. 5b. Somewhat surprisingly, our cause checker based on cause synthesis performs significantly better on all benchmarks. This shows that our characterization of causes as complements of the upward closure of the negated effect (cf. Section 5.1) is more efficient than encoding the cause-checking instances into a hyperlogic, as done by CATS.

7 Related Work

The study of causality and its applications in formal methods has gained great interest in recent years [3]. In a finite setting, Ibrahim et al. use SAT solvers and linear programming to check [35] and infer [34] actual causes. Our definition of actual causality for reactive systems extends the definitions of Coenen et al. [19] to cases in which the limit assumption does not hold. While Coenen et al. study the theory of actual causality [29] in reactive systems, they do not provide a way to *generate* causes and explanations. In terms of cause *synthesis*, the most related work is by Beutner et al. [11], which checks causality and generates causes based on sketching. Unlike ours, their tool is only applicable for the small fragment of LTL containing only \bigcirc operators, while we are able to generate temporal causes for all ω -regular specifications.

In a series of works, Leue et al. study symbolic description of counterfactual causes in *Event Order Logic* [14, 38, 39]. However, this logic can only reason about the ordering of events, and not their absolute timing, as we can do with ω -regular properties (e.g., specifying that the input at the second position is the cause).

Gössler and Métayer [24] define causality for component-based systems, and Gössler and Stefani [25] study causality based on counterfactual builders. Their formalisms differ from ours, which is based on Coenen et al. [19], and none of the works considers cause *synthesis*.

Most other works related to cause synthesis concern generating explanations for effects observed on finite traces [5, 26, 27, 49], or effects restricted to safety properties [43]. In the context of cause synthesis over infinite traces for effects given as temporal specifications, existing works are limited to causes given as sets of events (i.e., atomic propositions and times points) [7, 18, 32] or take a state-centric view to, e.g., measure the responsibility of a state for an observed effect property [1, 4, 42].

8 Conclusion

This paper presents the first complete algorithm to compute temporal causes for arbitrary ω -regular properties. It is based on a new, generalized version of temporal causality that solves a central dilemma of previous definitions by loosening the assumptions on similarity relations. From a philosophical perspective, this is an immense step forward since it is the first definition that accommodates the canonical similarity relation used in previous literature. Our experimental results show that our generalization also leads to significant improvements from a practical perspective. These mainly stem from characterizing causes based on set-closure properties, which may be an interesting approach for counterfactual causality in other formalisms. Besides, our work opens up exciting research directions on generating explanations from temporal causes, i.e., as formulas or annotations in highlighted counterexamples.

Acknowledgements. We thank Matthias Cosler and Frederik Schmitt for providing us with the neural synthesis benchmarks. This work was partially supported by the DFG in project 389792660 (Center for Perspicuous Systems, TRR 248) and by the ERC Grant HYPER (No. 101055412).

References

1. Baier, C., van den Bossche, R., Klüppelholz, S., Lehmann, J., Piribauer, J.: Backward responsibility in transition systems using general power indices. In: Wooldridge, M.J., Dy, J.G., Natarajan, S. (eds.) *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, 20-27 February 2024, Vancouver, Canada*, pp. 20320–20327. AAAI Press (2024). <https://doi.org/10.1609/AAAI.V38I18.30013>
2. Baier, C., Coenen, N., Finkbeiner, B., Funke, F., Jantsch, S., Siber, J.: Causality-based game solving. In: Silva, A., Leino, K.R.M. (eds.) *CAV 2021*. LNCS, vol. 12759, pp. 894–917. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81685-8_42
3. Baier, C., Dubslaff, C., Funke, F., Jantsch, S., Majumdar, R., Piribauer, J., Ziemek, R.: From Verification to Causality-Based Explications. In: Bansal, N., Merelli, E., Worrell, J. (eds.) *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Leibniz International Proceedings in Informatics (LIPIcs), vol. 198, pp. 1:1–1:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021). <https://drops.dagstuhl.de/opus/volltexte/2021/14070>
4. Baier, C., Dubslaff, C., Funke, F., Jantsch, S., Piribauer, J., Ziemek, R.: Operational causality - necessarily sufficient and sufficiently necessary. In: Jansen, N., Stoelinga, M., van den Bos, P. (eds.) *A Journey from Process Algebra via Timed Automata to Model Learning - Essays Dedicated to Frits Vaandrager on the Occasion of His 60th Birthday*. Lecture Notes in Computer Science, vol. 13560, pp. 27–45. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15629-8_2
5. Ball, T., Naik, M., Rajamani, S.K.: From symptom to cause: localizing errors in counterexample traces. In: Aiken, A., Morrisett, G. (eds.) *Conference Record of POPL 2003: The 30th SIGPLAN-SIGACT Symposium on Principles of Programming Languages, New Orleans, Louisiana, USA, 15-17 January 2003*, pp. 97–105. ACM (2003). <https://doi.org/10.1145/604131.604140>
6. Baumeister, J., Coenen, N., Bonakdarpour, B., Finkbeiner, B., Sánchez, C.: A temporal logic for asynchronous hyperproperties. In: Silva, A., Leino, K.R.M. (eds.) *CAV 2021*. LNCS, vol. 12759, pp. 694–717. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81685-8_33
7. Beer, I., Ben-David, S., Chockler, H., Orni, A., Trefler, R.: Explaining counterexamples using causality. In: Bouajjani, A., Maler, O. (eds.) *CAV 2009*. LNCS, vol. 5643, pp. 94–108. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02658-4_11
8. Beutner, R., Finkbeiner, B.: HyperATL^{*}: A logic for hyperproperties in multi-agent systems. *Log. Methods Comput. Sci.* **19**, 13:1–13:44 (2023)
9. Beutner, R., Finkbeiner, B.: Model checking omega-regular hyperproperties with autohyperq. In: Piskac, R., Voronkov, A. (eds.) *LPAR 2023: Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and*

- Reasoning, Manizales, Colombia, 4-9th June 2023. EPiC Series in Computing, vol. 94, pp. 23–35. EasyChair (2023). <https://doi.org/10.29007/1XJT>
10. Beutner, R., Finkbeiner, B., Frenkel, H., Metzger, N.: Second-order hyperproperties. In: Enea, C., Lal, A. (eds.) Computer Aided Verification - 35th International Conference, CAV 2023, Paris, France, 17-22 July 2023, Proceedings, Part II. Lecture Notes in Computer Science, vol. 13965, pp. 309–332. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37703-7_15
 11. Beutner, R., Finkbeiner, B., Frenkel, H., Siber, J.: Checking and sketching causes on temporal sequences. In: André, É., Sun, J. (eds.) Automated Technology for Verification and Analysis - 21st International Symposium, ATVA 2023, Singapore, 24-27 October 2023, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14216, pp. 314–327. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-45332-8_18
 12. Biere, A.: The AIGER And-Inverter Graph (AIG) format version 20071012. Tech. Rep. 07/1, Inst. f. Form. Model. u. Verifikation, Johannes Kepler University (2007)
 13. Buechi, J.R.: On a decision method in restricted second-order arithmetic. In: International Congress on Logic, Methodology, and Philosophy of Science (1962)
 14. Caltais, G., Guetlein, S.L., Leue, S.: Causality for general LTL-definable properties. In: Workshop on Formal Reasoning About Causation, Responsibility, and Explanations in Science and Technology, CREST 2018. EPTCS, vol. 286 (2018). <https://doi.org/10.4204/EPTCS.286.1>
 15. Chaki, S., Groce, A., Strichman, O.: Explaining abstract counterexamples. In: Taylor, R.N., Dwyer, M.B. (eds.) Proceedings of the 12th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2004, Newport Beach, CA, USA, October 31 - November 6, 2004, pp. 73–82. ACM (2004). <https://doi.org/10.1145/1029894.1029908>
 16. Clarkson, M.R., Finkbeiner, B., Koleini, M., Micinski, K.K., Rabe, M.N., Sánchez, C.: Temporal logics for hyperproperties. In: Abadi, M., Kremer, S. (eds.) POST 2014. LNCS, vol. 8414, pp. 265–284. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54792-8_15
 17. Clarkson, M.R., Schneider, F.B.: Hyperproperties. *J. Comput. Secur.* **18**(6), 1157–1210 (2010). <https://doi.org/10.3233/JCS-2009-0393>
 18. Coenen, N., Dachsel, R., Finkbeiner, B., Frenkel, H., Hahn, C., Horak, T., Metzger, N., Siber, J.: Explaining hyperproperty violations. In: Shoham, S., Vizel, Y. (eds.) Computer Aided Verification - 34th International Conference, CAV 2022, Haifa, Israel, August 7-10, 2022, Proceedings, Part I. LNCS, vol. 13371, pp. 407–429. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-13185-1_20
 19. Coenen, N., Finkbeiner, B., Frenkel, H., Hahn, C., Metzger, N., Siber, J.: Temporal causality in reactive systems. In: Bouajjani, A., Holík, L., Wu, Z. (eds.) Automated Technology for Verification and Analysis - 20th International Symposium, ATVA 2022, Virtual Event, 25-28 October 2022, Proceedings. LNCS, vol. 13505, pp. 208–224. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19992-9_13
 20. Cosler, M., Schmitt, F., Hahn, C., Finkbeiner, B.: Iterative circuit repair against formal specifications. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, 1-5 May 2023. OpenReview.net (2023). <https://openreview.net/pdf?id=SEcSahl0Ql>
 21. Duret-Lutz, A., et al.: From spot 2.0 to spot 2.10: What’s new? In: Shoham, S., Vizel, Y. (eds.) Computer Aided Verification - 34th International Conference, CAV 2022, Haifa, Israel, 7-10 August 2022, Proceedings, Part II. Lecture Notes in Computer Science, vol. 13372, pp. 174–187. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-13188-2_9

22. Finkbeiner, B., Frenkel, H., Metzger, N., Siber, J.: Synthesis of temporal causality. CoRR (2024). <https://doi.org/10.48550/ARXIV.2405.10912>, <https://arxiv.org/abs/2405.10912>, full version
23. Finkbeiner, B., Siber, J.: Counterfactuals modulo temporal logics. In: Piskac, R., Voronkov, A. (eds.) LPAR 2023: Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Manizales, Colombia, 4-9th June 2023. EPiC Series in Computing, vol. 94, pp. 181–204. EasyChair (2023). <https://doi.org/10.29007/QTW7>
24. Gössler, G., Le Métayer, D.: A general trace-based framework of logical causality. In: Fiadeiro, J.L., Liu, Z., Xue, J. (eds.) FACS 2013. LNCS, vol. 8348, pp. 157–173. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07602-7_11
25. Gössler, G., Stefani, J.: Causality analysis and fault ascription in component-based systems. Theor. Comput. Sci. **837** (2020). <https://doi.org/10.1016/j.tcs.2020.06.010>
26. Groce, A., Chaki, S., Kroening, D., Strichman, O.: Error explanation with distance metrics. Int. J. Softw. Tools Technol. Transf. **8**(3) (2006). <https://doi.org/10.1007/s10009-005-0202-0>
27. Groce, A., Kroening, D., Lerda, F.: Understanding counterexamples with explain. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, pp. 453–456. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27813-9_35
28. Gutsfeld, J.O., Müller-Olm, M., Ohrem, C.: Propositional dynamic logic for hyperproperties. In: International Conference on Concurrency Theory, CONCUR 2020. LIPIcs, vol. 171. Schloss Dagstuhl (2020). <https://doi.org/10.4230/LIPIcs.CONCUR.2020.50>
29. Halpern, J.Y.: A modification of the Halpern-pearl definition of causality. In: Yang, Q., Wooldridge, M.J. (eds.) Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, 25-31 July 2015, pp. 3022–3033. AAAI Press (2015). <http://ijcai.org/Abstract/15/427>
30. Halpern, J.Y.: Actual Causality. MIT Press (2016)
31. Halpern, J.Y., Pearl, J.: Causes and explanations: a structural-model approach: Part 1: Causes. In: Breese, J.S., Koller, D. (eds.) UAI 2001: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, 2-5 August 2001, pp. 194–202. Morgan Kaufmann (2001)
32. Horak, T., et al.: Visual analysis of hyperproperties for understanding model checking results. IEEE Trans. Vis. Comput. Graph. **28**(1) (2022) <https://doi.org/10.1109/TVCG.2021.3114866>
33. Hume, D.: An Enquiry Concerning Human Understanding. London (1748)
34. Ibrahim, A., Pretschner, A.: From checking to inference: actual causality computations as optimization problems. In: Hung, D.V., Sokolsky, O. (eds.) ATVA 2020. LNCS, vol. 12302, pp. 343–359. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59152-6_19
35. Ibrahim, A., Rehwald, S., Pretschner, A.: Efficiently checking actual causality with sat solving. Eng. Secure Dependable Softw. Syst. **53**, 241–255 (2019)
36. Kupriyanov, A., Finkbeiner, B.: Causality-based verification of multi-threaded programs. In: D’Argenio, P.R., Melgratti, H. (eds.) CONCUR 2013. LNCS, vol. 8052, pp. 257–272. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40184-8_19
37. Kupriyanov, A., Finkbeiner, B.: Causal termination of multi-threaded programs. In: Biere, A., Bloem, R. (eds.) CAV 2014. LNCS, vol. 8559, pp. 814–830. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08867-9_54

38. Leitner-Fischer, F., Leue, S.: Causality checking for complex system models. In: Giacobazzi, R., Berdine, J., Mastroeni, I. (eds.) VMCAI 2013. LNCS, vol. 7737, pp. 248–267. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35873-9_16
39. Leitner-Fischer, F., Leue, S.: SpinCause: a tool for causality checking. In: International Symposium on Model Checking of Software, SPIN 2014. ACM (2014). <https://doi.org/10.1145/2632362.2632371>
40. Lewis, D.K.: Causation. *J. Philos.* **70**(17), 556–567 (1973). <https://doi.org/10.2307/2025310>
41. Lewis, D.K.: Counterfactuals. Blackwell, Cambridge, MA, USA (1973)
42. Mascle, C., Baier, C., Funke, F., Jantsch, S., Kiefer, S.: Responsibility and verification: importance value in temporal logics. In: 36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021, pp. 1–14. IEEE (2021). <https://doi.org/10.1109/LICS52264.2021.9470597>
43. Parreaux, J., Piribauer, J., Baier, C.: Counterfactual causality for reachability and safety based on distance functions. In: Achilleos, A., Monica, D.D. (eds.) Proceedings of the Fourteenth International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2023, Udine, Italy, 18-20 September 2023. EPTCS, vol. 390, pp. 132–149 (2023). <https://doi.org/10.4204/EPTCS.390.9>
44. Pnueli, A.: The temporal logic of programs. In: 18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977, pp. 46–57. IEEE Computer Society (1977). <https://doi.org/10.1109/SFCS.1977.32>
45. Pnueli, A., Rosner, R.: On the synthesis of a reactive module. In: Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, 11-13 January 1989, pp. 179–190. ACM Press (1989). <https://doi.org/10.1145/75277.75293>
46. Sallinger, S., Weissenbacher, G., Zuleger, F.: A formalization of heisenbugs and their causes. In: Ferreira, C., Willemse, T.A.C. (eds.) Software Engineering and Formal Methods - 21st International Conference, SEFM 2023, Eindhoven, The Netherlands, 6-10 November 2023, Proceedings. Lecture Notes in Computer Science, vol. 14323, pp. 282–300. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-47115-5_16
47. Schewe, S.: Büchi complementation made tight. In: Albers, S., Marion, J. (eds.) 26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, 26-28 February 2009, Freiburg, Germany, Proceedings. LIPIcs, vol. 3, pp. 661–672. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany (2009). <https://doi.org/10.4230/LIPICs.STACS.2009.1854>
48. Schmitt, F., Hahn, C., Rabe, M.N., Finkbeiner, B.: Neural circuit synthesis from specification patterns. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, 6-14 December 2021, virtual, pp. 15408–15420 (2021). <https://proceedings.neurips.cc/paper/2021/hash/8230bea7d54bcd99c9cfe85cb07313d5-Abstract.html>
49. Wang, C., Yang, Z., Ivančić, F., Gupta, A.: Whodunit? Causal analysis for counterexamples. In: Graf, S., Zhang, W. (eds.) ATVA 2006. LNCS, vol. 4218, pp. 82–95. Springer, Heidelberg (2006). https://doi.org/10.1007/11901914_9

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

