

From Verification to Causality-Based Explications

Christel Baier  

Technische Universität Dresden, Germany

Florian Funke  

Technische Universität Dresden, Germany

Rupak Majumdar  

MPI-SWS, Kaiserslautern, Germany

Robin Ziemek  

Technische Universität Dresden, Germany

Clemens Dubslaff  

Technische Universität Dresden, Germany

Simon Jantsch  

Technische Universität Dresden, Germany

Jakob Piribauer  

Technische Universität Dresden, Germany

Abstract

In view of the growing complexity of modern software architectures, formal models are increasingly used to understand *why* a system works the way it does, opposed to simply verifying *that* it behaves as intended. This paper surveys approaches to formally explicate the observable behavior of reactive systems. We describe how Halpern and Pearl’s notion of actual causation inspired verification-oriented studies of cause-effect relationships in the evolution of a system. A second focus lies on applications of the Shapley value to responsibility ascriptions, aimed to measure the influence of an event on an observable effect. Finally, formal approaches to probabilistic causation are collected and connected, and their relevance to the understanding of probabilistic systems is discussed.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases Model Checking, Causality, Responsibility, Counterfactuals, Shapley value

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.1

Category Invited Talk

Funding This work was funded by DFG grant 389792660 as part of TRR 248 – CPEC (see <https://perspicuous-computing.science>), the Cluster of Excellence EXC 2050/1 (CeTI, project ID 390696704, as part of Germany’s Excellence Strategy), DFG-projects BA-1679/11-1 and BA-1679/12-1, and the European Research Council under the Grant Agreement 610150 (<http://www.impact-erc.eu/>) (ERC Synergy Grant ImPACT).

1 Introduction

Modern software systems are increasingly complex and even small changes to a system or its environment may lead to unforeseen and disastrous behaviors. As software controls more aspects of our lives everyday, it is desirable – and for widespread acceptance in societal decisions, eventually inevitable – to have comprehensive and powerful techniques available to understand what a system does.

The field of formal methods has developed a portfolio of tools that provide confidence in the working of complex software systems. In formal methods, one builds a formal model of a system and specifies its desired behavior in an appropriate (temporal) logical formalism. Algorithmic techniques such as model checking [12, 31] can answer the question whether the model satisfies the specification, or in other words, whether the system behaves as intended, often in a “push button” way. Moreover, an important aspect of these algorithms is that they can produce independently verifiable justifications of their outcome, such as *counterexamples* or *certificates* to justify the violation or correctness of a property, respectively. Since the earliest successes of model checking, the availability of counterexample traces was stated as a major advantage for the method over deductive verification [30]. As model checkers became



© Christel Baier, Clemens Dubslaff, Florian Funke, Simon Jantsch, Rupak Majumdar, Jakob Piribauer, and Robin Ziemek;

licensed under Creative Commons License CC-BY 4.0

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).

Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 1; pp. 1:1–1:20

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



more complex, concerns about their correct implementation led to research on producing certificates for correctness. Examples are inductive invariants or derivations in a deductive system [62, 64, 44, 70] that can be checked independently from the verification process.

While certificates and counterexample traces can provide a useful explication about the behavior of a system, they only provide rudimentary understanding of *why* a system works the way it does. In epistemic terms, the outcome of model checking applied to a system and a specification provides *knowledge that* a system satisfies a specification or not in terms of an assertion (whether the system satisfies the specification) and a justification (certificate or counterexample) to increase the belief in the result. However, model checking usually does not provide *understanding on why* a system behaves in a certain way. Such an understanding can be obtained by causal links between possible events and their observed outcome.¹

The need to better understand why a system is correct or incorrect has led to a broad research program on models and reasoning methods that aim to provide such knowledge of causes (see, e.g., [97, 98]). The goal of this survey is to summarize research on causal reasoning in the field of verification and highlight the challenges that lie ahead.

The first step in understanding knowledge of causes is the mathematical formulation and study of *causality*. Grasping the intuitive concept of cause-effect relationships in a formal model has proved notoriously difficult. Centuries of philosophical reasoning on the subject have distilled the *counterfactuality principle* [67, 68, 88] as a central feature of what constitutes an actual cause: if the cause had not occurred, then the effect would not have happened. While the counterfactuality principle was generally agreed upon, a rigorous mathematical formulation was developed only recently, through the seminal work of Halpern and Pearl and their coworkers [95, 45, 46, 55]. In a nutshell, they model causal systems using *structural equation models*, and provide a set of axioms to characterize when an event is an actual cause of another. We provide a summary of the foundations of causality and some of their applications in verification in Section 2.

While causality is a qualitative concept, in that an event is an actual cause of another or is not, more recent work considers *quantitative* measures of *responsibility*. Responsibility measures the relative importance that an event had in causing another event. In other words, the responsibility of an acting agent gauges what fraction of an observable effect can be attributed to that agent's behavior. Here, an agent could be, e.g., an individual, a coalition, a software component, or device in a computer network. Chockler and Halpern [23] define the *degree of responsibility* of an actual cause in the Halpern-Pearl sense based on the cardinality of the smallest witness change that makes an event a cause of another. A more recent strand for formalizing responsibility is based on the *Shapley value* [106]. In cooperative game theory, the Shapley value measures the influence of an agent on the outcome jointly brought about by the agents and is classically used to find a fair division of a cost or a surplus among them. The appeal of the Shapley value stems on the one hand from its uniqueness with respect to a relatively simple set of axioms, and on the other hand from its seemingly universal applicability. Research on employing Shapley-like values for the explication of machine-learning predictions [117, 35, 91, 1, 110] and the behavior of formal models [119, 11, 93, 10] is currently very active. A summary of applications of Shapley values in the verification context is provided in Section 3.

¹ Epistemologists since Gettier [47] will recognize that justified true belief does not constitute a comprehensive theory of knowledge. For the same reason, a theory of understanding as knowledge of causes is a matter of vigorous debate, with Gettier-like counterexamples [52, 100, 101]. These subtle epistemic issues are orthogonal to our work.

From a systematic viewpoint, causality and responsibility can be understood in either a *backward* or a *forward* manner [113]. In the backward or *ex post* setting, an effect has already transpired, and the goal is to describe its causes and determine their relative influence in producing the effect. The actual causation framework by Halpern and Pearl follows this paradigm, and therefore also most approaches presented in Section 2. In the forward or *ex ante* setting, a reasoning model includes possible contingencies and the goal is to characterize the global power of agents and events in affecting the outcome. The forward responsibility in game structures (see Section 3.1) and the importance value for temporal logics (see Section 3.2) pursue this pattern. There are also attempts to express forward-looking causality notions by structural equations in the context of *accountability* [73]. Seen from an operational angle, the distinction between backward and forward notions loosely relates to when the causal analysis is executed. Backward notions tend to be applicable at inspection time, e.g., to guide the debugging process in post mortem analyses. Forward notions are prone to be used at design time of the system, laying out general phenomena of its inner workings.

Finally, we consider causality in the setting of probabilistic models. Unlike the deterministic setting, mathematical notions of causality and responsibility are less understood. There is widespread agreement in the philosophy literature that a quintessential characteristic of causes in the probabilistic setting is the *probability-raising property* [111, 21, 107, 39]: an occurrence of the cause should increase the probability of subsequently observing the effect. Nevertheless, it has also been observed that simply taking probabilities often leads to counterintuitive phenomena owing to mutual dependencies and latent correlations with other events [104, 21, 107]. Section 4 discusses attempts to formulate probability-raising approaches to causation for operational probabilistic models. These approaches tend to produce forward notions since probabilities inherently refer to a collection of evolutions in which an event happened. There are numerous philosophical accounts on actual probabilistic causation [89, 84, 42]. In terms of formal models, Pearl's early notion of actual causality in terms of *causal beams* [96, 97] entailed probabilistic flavor, and the *causal probabilistic logic* of [115, 14] describes a language for reasoning about probabilistic causation. Nonetheless, we are aware of only a few works that study a probabilistic version of causality in operational models (see, e.g., [75, 2, 37, 9]). Along these lines, we point out open directions for research that focus on the operational point of view.

2 Counterfactual Notions of Causality

An important starting point for the study of causality in formal methods is the influential work by Halpern and Pearl [58, 59, 60, 56, 57] on *actual causality*, henceforth abbreviated *HP causality*. We provide a brief and informal overview of their definition.

Halpern and Pearl use *structural equation systems* as a modeling language for causal models. A causal model relies on exogenous variables U and endogenous variables V , representing external or independent factors and internal factors, respectively. The value of each endogenous variable $x \in V$ is specified by a deterministic function f_x that may depend on exogenous variables and on endogenous variables that are preceding x with respect to a fixed order on V . Intuitively, a causal model can be thought of as an arithmetic circuit whose primary inputs are the exogenous variables and where some of whose internal nodes are labeled by the endogenous variables. The circuit then specifies the functions defining the endogenous variables as well as the dependencies between the variables.

More formally, let $M = (U, V, \{f_x\}_{x \in V})$ be a causal model. Given a formula φ over the exogenous and endogenous variables (in some appropriate logic), and a *context* \vec{u} that assigns values to all variables in U , the goal of actual causality is to state whether an assignment of values $\vec{X} = \vec{x}$ to a subset $X \subseteq V$ is a cause of φ . Halpern and Pearl define $\vec{X} = \vec{x}$ to be a *cause* of a formula φ in (M, \vec{u}) if the following three axioms hold.

- AC1** both the cause and the effect are true: the model (M, \vec{u}) satisfies $\vec{X} = \vec{x}$ as well as φ ,
- AC2** the principle of counterfactual dependence (discussed below), and
- AC3** causes are minimal: no partial assignment of $\vec{X} = \vec{x}$ satisfies AC1 and AC2.

The key to AC2 is captured by the notion of *interventions*, describing a direct assignment of values to some endogenous variables while disregarding their defining functions. Formally, $[\vec{Y} \leftarrow \vec{y}]$ stands for the intervention on variables $Y \subseteq V$ by assigning them values \vec{y} and leaving all other values for variables $V \setminus Y$ to follow from their defining functions. Then, $[\vec{Y} \leftarrow \vec{y}]\psi$ describes the impact of an intervention on a formula ψ . An intervention thus can represent a counterfactual: *what if* variables in Y took values \vec{y} instead of their actual values? Turning back to the definition of actual causes for φ , axiom AC2 now requires the existence of an intervention $[\vec{X} \leftarrow \vec{x}']$ on the variables in X such that the effect φ is not observable, i.e., $[\vec{X} \leftarrow \vec{x}']\neg\varphi$ holds in (M, \vec{u}) . The precise definition of AC2 is, however, more involved and several variants exist for AC2 to account for different settings and applications.²

2.1 Instances of HP Causality in Verification

Principles of causality have been used, often implicitly, in formal verification for a long time. An early example is *program slicing* (see, e.g., [61]) where by following program dependencies one aims to identify approximations of an actual cause for reaching a program location. Causality is also a key concept in error localization, the problem of reducing a counterexample trace for ease of debugging [120, 13, 105, 53, 118, 72, 121]. A correspondence of causality in counterexample traces to finding minimal UNSAT cores has been identified in [16]. Early and influential work on causality in formal verification is exemplified by research on *vacuity* and *coverage*. Vacuity [17, 81, 102] explicates whether a positive verification result originates from an unintended trivial behavior. Coverage [65, 25, 27, 26] is dual to vacuity, and explicates whether certain parts of the system were not relevant for the successful result. While for determining vacuity one considers small changes to the specification and checks whether these change the result, coverage is obtained by perturbations to the system rather than the specification and is actually a particular instance of HP causality.

Temporal logics play a crucial role in the verification context to describe properties of and requirements on the system. Common temporal logics are, e.g., *computation tree logic* (CTL) [29] or *linear temporal logic* (LTL) [99]. In LTL, e.g., $\neg E \mathcal{U} C$ describes that an effect E does not occur before a cause C and $\diamond E$ stands for the effect E to eventually occur.

From Coverage to HP Causality. Coverage itself is a concept with a manifold of incarnations and we focus here on the formalization by [24], where the connection of coverage to HP causality has been addressed. The operational model is provided by a Kripke structure K , i.e., a finite directed graph over states labeled by atomic propositions. Further, we are given an atomic proposition q and a specification ψ expressed in an appropriate (temporal)

² Halpern and Pearl's definition of causality underwent a considerable amount of development over the past 20 years, primarily varying AC2. One usually distinguishes the *original* version [58], the *updated* version [59, 60], and the *modified* version [56, 57] (alongside variations by other authors [54, 63]).

logic over the set of atomic propositions such that K satisfies ψ . Then, a state s of K is said to be q -covered if changing the truth value of q in s leads to a structure that does not satisfy ψ . Considering the hypothetical mutant system in which q takes the opposite value in s corresponds to a counterfactual notion from the causality literature. Yet, coverage only allows simple counterfactuals containing *individual* changes to the system. As pointed out in [24], it is for this reason that coverage at times fails to express deeper dependencies involved in the satisfaction of ψ .

To define a *cause* in this setting, one can consider the following simple causal model: for each state s , there is one endogeneous variable v_s , which specifies whether the value of q in s is swapped in contrast to the original structure K , or not. One first refers to a context where all variables v_s are set to **false** and then considers possible swap operations. From K and ψ one can derive a Boolean function φ over the endogeneous variables V such that an instantiation $I: V \rightarrow \{\mathbf{true}, \mathbf{false}\}$ satisfies φ if and only if swapping the truth value of q exactly in states s with $I(v_s) = \mathbf{true}$ leads to a structure satisfying ψ . Now s is a *cause of ψ with respect to q* [24, Definition 3.2] if there exists a set of variables Y such that $[Y \leftarrow \mathbf{true}]\varphi$ and $[Y \cup \{v_s\} \leftarrow \mathbf{true}]\neg\varphi$ hold. In other words, s is a cause if there exists a set of states S' (corresponding to variables Y) such that swapping q in S' leads to a structure satisfying ψ ³, but swapping the value of q in S' and s gives a structure falsifying ψ . These two conditions postulate precisely the axiom AC2, which takes a simpler form than usual thanks to the lack of higher-order dependencies among the variables in this causal model. In the presented causal model, axiom AC1 holds by the assumption that K satisfies ψ , and the minimality axiom AC3 is trivially fulfilled as only single states are considered as potential causes.

While this causal model is very simple, in particular it does not include any dependencies in between variables, the work in [24] shows that even such restricted models are useful.

Fault Localization. The causality interpretation of coverage presented above takes a forward-looking perspective in that changes to the system are globally tested against the given specification. In [16], a similar approach is applied to the backward-oriented setting of fault localization, i.e., the problem of pointing out those parts of a (finite) counterexample trace π that are most relevant for violation of a given linear-time specification φ . In this incarnation of HP causality, the endogeneous variables V contain a variable $v_{(s,q)}$ for each pair consisting of a state s and atomic proposition q of the Kripke structure. These variables can take values $\{\mathbf{true}, \mathbf{false}\}$ and the interpretation is exactly as before, namely $v_{(s,q)} = \mathbf{true}$ means that the truth value of q in state s is changed in contrast to the initial context. Moreover, the axiom AC2 takes the same form as in the previous case. Specifically, it expresses that there is a set of variables $Y \subseteq V$ such that changing the truth value for the corresponding state-proposition pairs lets π still violate φ , while additionally swapping q in s leads to π satisfying φ (interpreted over a weak semantics of LTL on finite paths).

Counterfactual Reasoning for Configurable Systems. Nowadays, almost every practical software system is configurable, let it be using `#ifdef` constraints or through *features* [5]. Features inherently have a designated meaning, usually expressed by their name, e.g., a “verbose” feature indicates that the software will expose additional information during runtime. Debugging configurable systems is challenging, as the number of possible systems suffers from an exponential blowup in the number of features. While there are specifically tailored methods for analyzing configurable systems [112], e.g., through model checking [32, 38],

³ The updated version of HP causality [59, 60] would require this condition also for all subsets of S' .

research on identifying root causes in configurable systems on the abstraction level of features is still in its infancy. Such a causal analysis can provide useful insights for debugging: developers can focus on the parts implementing the features identified to be responsible for the bug, and users can obtain suggestions to reconfigure the system to not expose the bug. First ideas to explicate which feature activations and deactivations cause an effect in configurable systems were described in [6]. There, the set of feature configurations with observable effect is obtained by configurable systems analysis, e.g., through family-based verification [32, 112, 38, 28]. Exploiting the Boolean case of HP causality [40, 69], those partial feature configurations can be determined where the corresponding systems all show the effect (see AC1), for which there is a reconfiguration that does not exhibit the effect (AC2), and that are minimal (AC3).

2.2 Further Approaches Inspired by HP Causality

The work [34] presents a formal definition of actual causes in the setting of concurrent interacting programs. Originating from logs written by the concurrent system, the goal is to localize causes in those *program actions* that are most relevant for the violation of a desirable property. The approach is investigated in detail for the prominent class of safety properties, with a view towards legal accountability in security-critical systems.

In [78], a causality-based approach to explain *timed diagnostic traces* has been presented, which are used as counterexamples for model-checking results in timed systems. Such traces represent a set of violating executions and the goal of [78] is to compute the parts that can be considered causal for violating the property.

A different definition inspired by HP causality was used in [86]. There, causes for reachability properties are formulas of a temporal logic called *event order logic*, used to describe temporal relations between events. Algorithms to compute causes in this sense were also studied in [15, 77], and the approach was extended to handle general LTL formulas as effects rather than just reachability in [20].

In [49, 50] the authors argue that the HP causality, which is propositional in nature, is not the ideal starting point for a framework of causality in formal verification. They present a formalism which is based on counterfactual reasoning, uses system traces as first-class objects and is designed to work for compositional systems. In [51] the formalism is further generalized by defining abstract *counterfactual builders*, which specify what alternative scenarios should be considered for counterfactual reasoning. Further, [51] also considers hyperproperties as specifications. While hyperproperties are useful to specifying system properties, it was observed in [33] that they can also be used to formalize causality. Similar observations have been made for probabilistic causation [2, 37].

3 Responsibility and Shapley-like Ascriptions

While the previous section defined and applied qualitative concepts of causation, this section shifts the focus towards quantitative approaches of *responsibility*. Loosely speaking, responsibility refers to a numerical value designed to measure how much weight an event had in producing an effect, relative to concurring or competing events linked to the same effect. There is widespread agreement that a necessary condition for assuming responsibility is causal relevance of the event in question to the effect [41, 18]. As a consequence, the term *responsibility* usually builds directly or indirectly on concepts of causality. While the specific numerical value in a notion of responsibility may not have a semantic content, it can order the events in terms of their causal relevance.

Chockler and Halpern [23] introduced the notion of *degree of responsibility*, which is attributed to actual causes in causal models of HP causality. This degree measures how many changes to the evolution of events are necessary until counterfactual values for the actual cause change the observable effect. In [24] this notion is combined with the study of mutant coverage to build a degree of responsibility in CTL model checking assigned to state-proposition pairs (see also Section 2.1).

The degree of responsibility measures the influence of an event by looking at how many further counterfactual changes are (minimally) required to swap the effect, but it does not take into account how many such minimal sets of changes exist. One can argue that a cause is individually more influential if it admits *many* such sets since this means less dependencies on other events. This rationale has generated an active strand in formalization of responsibility based on the *Shapley value* [106]. The Shapley value is a central solution concept from theoretical economics and was originally designed to find a fair distribution of a financial surplus that was brought about cooperatively by a number of producers.

Formally, a *cooperative game* with n players is a mapping $g: 2^{[n]} \rightarrow \mathbb{R}$ such that $g(\emptyset) = 0$, where $[n] = \{1, \dots, n\}$. The value $g(C)$ is meant to represent the surplus (or, depending on the specific situation, the cost) that the coalition $C \subseteq [n]$ can ensure upon acting collaboratively. The Shapley value of player i is then defined as

$$\text{Sh}(i) = \frac{1}{n!} \cdot \sum_{\pi \in S_n} g(\pi_{\geq i}) - g(\pi_{\geq i} \setminus \{i\}) \quad (1)$$

where S_n denotes the set of self-bijections $[n] \rightarrow [n]$ and where $\pi_{\geq i} = \{j \in [n] \mid \pi(j) \geq \pi(i)\}$ for a given $\pi \in S_n$. Intuitively, $g(\pi_{\geq i}) - g(\pi_{\geq i} \setminus \{i\})$ describes the marginal contribution of player i to the coalition $\pi_{\geq i}$. The Shapley value takes the average of all such marginal contributions. Thus, $\text{Sh}(i)$ is a measure for the overall influence of player i in the game g .

The general setup of cooperative games as real-valued functions on the power set of $[n]$ makes the Shapley value amenable to measuring the influence of abstract players in formalized situations of collaborative interaction. This rationale has recently been invoked for the interpretation of machine learning models [117, 35, 91, 1, 110]. In this case, the players are the input parameters to a machine learning model, and the Shapley value has the goal to measure the influence of each parameter on the output of the model.

This section outlines three approaches that employ the Shapley value as a means to distribute an overall effect into individual responsibilities. In Section 3.1 the general setting of an *extensive form game* is chosen and responsibilities are attributed to its players with respect to producing a certain outcome. Section 3.2 discusses a notion of importance of states for the satisfaction of an LTL property in a Kripke structure. Section 3.3 finally presents an extension of the Shapley value that can be used to define responsibilities in a setting of continuously varying parameters.

3.1 Responsibility in Game Structures

As summarized in Section 2, causal models are by now a fundamental building block for notions of actual causation in the verification domain. However, in complex scenarios that involve cooperative interaction, non-cooperative competition, and imperfect information, they fall short of modeling various natural features such as temporal sequentiality, knowledge, and agency. The work [10] presents an approach to establish notions of responsibility in these strategic settings by passing to *extensive form games* [116, 80]. These provide a popular formalism for studying the dynamics that underlie strategic interaction in the presence of

competing objectives. In a nutshell, an extensive form game is an explicit presentation of the strategic scenario in terms of a tree structure whose edges describe the transitions between states when actions are taken, certain states may be indistinguishable for the players given their knowledge, and each path from the root to a leaf is associated with an outcome. Apart from being a highly expressive model, a century of research on the subject has generated a rich set of solution concepts on which a study of responsibility can build, primarily following economic rationales.

In [10] three responsibility notions are defined with respect to an event E that is encoded by a binary labelling on the leaves of the game tree, i.e., E took place on a play or not. All three notions follow the common two-step process consisting of first defining (qualitatively) what it means for a coalition C to be responsible and then extracting (quantitatively) an individual responsibility value through an application of the Shapley value on coalition responsibilities. That is, one takes the cooperative game g to take binary values $\{0, 1\}$ depending on whether or not a coalition is responsible. They also share the counterfactual paradigm in that a necessary condition for being responsible for the occurrence of E is the power to preclude E . While the notions can be ordered according to their logical strength, they are perhaps best explained along two lines of distinction given by the *temporal perspective* and *epistemic state*.

The temporal perspective can be either *forward-looking* or *backward-looking* [113]. For forward-looking notions one attains a prospective, *ex ante* viewpoint that studies the preclusive power for the game as a whole. The forward-looking notion put forth in [10] is called *forward responsibility* and requires the coalition to possess a strategy that globally avoids E . In contrast, backward-looking notions consider a specific play from a retrospective, *ex post* viewpoint and study who was responsible for E as the play evolved.

Depending on how the epistemic state is taken into account, [10] distinguishes *strategic* backward responsibility and *causal* backward responsibility. In order for a coalition to be strategically backward responsible, it must have had the power to avoid E at some point on the play and it must have been aware of this fact *given its epistemic knowledge*. In situations of imperfect information, this latter condition is crucial for arriving at a *responsibility-as-capacity* notion [113] in a strategic sense that goes beyond a mere counterfactuality check: when one does not know all relevant information, one can even bring about E inadvertently or unintendedly. Causal backward responsibility essentially drops the latter requirement in that the coalition is able to avoid E from some point on, everything else held fixed. This corresponds to the *responsibility-as-cause* from the classification presented in [113].

There is a translation of a causal model into an extensive form game under which causal backward responsibility corresponds exactly to but-for causes [10]. It can therefore happen that a player is causally backward responsible without belonging to an actual cause in the HP causality sense [59] and, therefore, with degree of responsibility 0 in the sense of [23]. In the prototypical example in which Suzy and Billy both throw rocks at a bottle and Suzy's stone hits first, Billy's degree of responsibility is 0, while both are attributed causal backward responsibility $1/2$. Since both players acted in exactly the same way based on the same information, there is reason to favor the latter symmetric notion, and avoid actual causation *en route* to accurately model intuitive responsibility concepts. A detailed comparison to causal models, other notions of responsibility in strategic games of imperfect information [119], and proof-theoretic approaches to formalize responsibility [19, 94] is given in [10].

3.2 The Importance Value for Temporal Logics

The paradigm of passing from binary coalitional responsibilities to quantitative individual responsibilities by virtue of the Shapley value is also applied in [93] to model check Kripke structures against temporal logic specifications. The resulting notion is called the *importance value* and measures the influence of a state in a system for the satisfaction of a given specification. Intuitively, a state is important in this framework if the way that the nondeterministic choices of the state are resolved has a large impact on whether the given specification is met.

Formally, let K be a Kripke structure with states S and a dedicated initial state, and φ be an LTL formula. Then one defines the cooperative game $g: 2^S \rightarrow \{0, 1\}$ using an induced two-player game as follows. For a set of states $C \subseteq S$ we let \mathcal{G}_C be the two-player game over the arena K where player SAT controls the states in C , player UNSAT controls the states in $S \setminus C$ and the winning condition is φ . Then, $g(C) = 1$ if player SAT wins \mathcal{G}_C , and $g(C) = 0$ otherwise. With this definition, the importance value $\mathcal{I}(s)$ of a state $s \in S$ with respect to K and φ is defined to be the Shapley value of player s in g (see Equation (1)). The notion can be straightforwardly extended to define the importance of a set of states $P_i \subseteq S$, where $S = P_1 \dot{\cup} \dots \dot{\cup} P_n$ is a given partition of the state space. This generalization is intended to take an existing compositional structure of the system appropriately into account.

The work [93] studies the associated computational problems of deciding whether $\mathcal{I}(s) > 0$ (called the *usefulness problem*) and deciding whether $\mathcal{I}(s) > \eta$ for a rational threshold η . The intrinsic complexity of solving two-player LTL-games (the decision problem is 2EXPTIME-complete) carries over to these problems. This computational intractability of the importance value motivates further studying the complexity when restricted to fragments of LTL, and tight complexity results were shown in [93] for a wide range of specifications.

In [93], the presented framework is also applied to CTL model checking of *modal transition systems* (MTS) [85]. MTSs have two levels of nondeterminism: the standard nondeterminism of the underlying graph and additionally a choice on which of the transitions in a state are actually included in the system. The latter kind of nondeterminism is used to design a two-player game where one player tries to satisfy the CTL specification and the other player tries to violate it. However, since the semantics of CTL relies on infinite trees and the order in which the branches are evolving has a strong impact on which player wins, there does not appear to be a natural candidate game that proceeds in a turn-by-turn fashion. Hence [93] considers *one-shot games* in which the players commit to a valid set of transitions in the states under their control once at the beginning of their play. This determines once more a binary cooperative game g that induces importance values in the same way as for LTL.

There is a straightforward generalization of the importance value to a $2\frac{1}{2}$ -player game \mathcal{G} in which the actions taken by the players are associated with probability distributions over the states. In this formalism, the players each make non-deterministic choices among its available actions, but the actual successor state then depends on a random choice according to the associated distribution. Given an LTL specification, the goal of SAT is to maximize the probability that the resulting path satisfies the specification, while UNSAT tries to minimize it. These $2\frac{1}{2}$ -player games are *determined* in a quantitative sense [92]: the maximal probability that can be enforced by SAT against all strategies of UNSAT is 1 minus the minimal probability that can be enforced by UNSAT against all strategies of SAT. This probability is called the *value* $\text{val}(\mathcal{G})$ of the game (see also the survey [22]). Let $S = S_{\text{SAT}} \dot{\cup} S_{\text{UNSAT}}$ be the partition of the states of \mathcal{G} into those under control of SAT and UNSAT, respectively. For a subset $C \subseteq S_{\text{SAT}}$ the value $g(C)$ is then defined as $\text{val}(\mathcal{G}_C)$, where \mathcal{G}_C is the $2\frac{1}{2}$ -player game obtained from \mathcal{G} by putting the states in $S \setminus C$ under the control of player UNSAT. Taking Shapley values as above then induces the importance value of a state in S_{SAT} .

3.3 Attributing Responsibility in Continuous Models

The Shapley value [106] is an inherently discrete solution concept. On the other hand, realistic formal models of reactive systems often entail continuous features such as timing [4, 36, 90], physical phenomena [3, 108], or parametric dependencies [71, 48, 79]. Notions of responsibility for these models therefore tend to require new mathematical approaches if the continuous nature is to be taken into account appropriately.

The continuous scenario seen from an economic angle generalizes (discrete) cooperative games: rather than just participating in a coalition, the n players of a game each pick a value v_i from a continuous domain $D_i \subseteq \mathbb{R}$ including 0 and the (generalized) game then determines a collective surplus or cost based on this input. This is formally described by a continuously differentiable function $g: D_1 \times \dots \times D_n \rightarrow \mathbb{R}$ such that $g(0, \dots, 0) = 0$. Economists usually take the domains to be of the form $D_i = [0, m_i)$ for some maximal input $m_i \in \mathbb{R}$, and further assume g to be non-decreasing with non-negative range. The *Aumann-Shapley value* [7] is a generalization of the Shapley value designed to provide a solution to the question how the value $g(v_1, \dots, v_n)$ should be “fairly” distributed among the players. It is one instance of what is called a *cost-sharing scheme* and admits an axiomatization in the spirit of its discrete predecessor [43, 110].

Inspired by this model, the work [11] presents an approach to measure the relative importance of the parameters on the behavior of *parametric Markov chains* for a wide range of properties, including ω -regular specifications, specifications in probabilistic CTL, and on expected rewards. Here, a parametric Markov chain is a directed graph where each edge is assigned a probability that may depend on a set of parameters such that for each instantiation of the parameters the probabilities outgoing from a state sum up to 1. A parametric Markov chain instantiated with fixed values for the parameters then coincides with a *discrete-time Markov chain* (DTMC). For this purpose the aforementioned assumptions on g must be relaxed: the continuously differentiable function $g: D \rightarrow \mathbb{R}$ has arbitrary domain $D \subseteq \mathbb{R}^n$ and is not subject to monotonicity and non-negativity restrictions. This also means that the canonical baseline value 0 for v_i is not always available anymore. The responsibility problem in this generalized setting then reads as follows: given g and two parameter choices $v, v' \in D$, how *responsible* is the i -th parameter for the observable change $g(v') - g(v)$? In this slightly generalized form, the Aumann-Shapley value of the i -th parameter is defined as

$$\text{AS}_i(g, v, v') = (v'_i - v_i) \cdot \int_0^1 \partial_i g(v + \alpha(v' - v)) d\alpha. \quad (2)$$

The integrand involves the i -th partial derivative of g and intuitively measures the marginal contribution of the i -th parameter at the points lying between v and v' . The integral then takes the average of these contributions along the straight line from v to v' . While taking the straight line is desirable in an economic context to meet the *average cost for homogeneous goods axiom* [43], this axiom is often void of meaning when applied to formal systems. When one replaces the straight line in Equation (2) with an arbitrary (monotonic) path from v to v' , then one speaks of *path attribution schemes* [109]. Of course, taking different paths induces different attributions, and which ones should be considered worthwhile depends on the specific scenario. This could for instance be due to potential restrictions on the way that changes on the parameters can be implemented in practice. The work [11] applies these path attribution schemes to the function induced by ω -regular or probabilistic CTL specifications on a parametric Markov chain. The set of axioms presented there is adjusted to this particular situation and justifies why one can conceive the value $\text{AS}_i(g, v, v')$ as the fraction of the observable effect $g(v') - g(v)$ that is produced by the i -th parameter.

It is noteworthy, however, that the approach put forth in [11] is by no means specific to the context of parametric Markov chains. Any scenario in which continuously varying parameters determine a value can in principle be handled similarly. Of course, which path attribution schemes should be regarded as meaningful needs to be checked case-by-case, and corresponding axiomatizations should be chosen with care. But it is no accident that the main decidability result in [11] is formulated in terms of path attribution schemes on functions in n independent variables – a generality that makes the approach potentially applicable for a range of similar problems.

4 Probabilistic Causation

As seen in the preceding sections, notions of causality and responsibility have been widely explored in the non-probabilistic setting. In contrast, there have been far less attempts at defining a suitable notion of causes for probabilistic operational systems such as Markov chains. However, probabilistic theories of causation have been considered in various philosophical accounts [111, 21, 107, 39]. One central idea behind these theories is the *probability-raising principle*, which goes back to Reichenbach [103, 104]. It states that causes should raise the probability of their effects. After observing a cause C , the probability of an effect E is higher than after observing that the cause has not occurred. Formulated with conditional probabilities, this can be written as

$$\Pr(E \mid C) > \Pr(E \mid \neg C), \quad \text{or equivalently} \quad \Pr(E \mid C) > \Pr(E).$$

For the conditional probabilities to be well-defined, it is necessary that $\Pr(C) > 0$ and $\Pr(\neg C) > 0$. Later on, we will make sure that the events conditioned on have positive probability. Note that if $\Pr(C) > 0$ and $\Pr(E \mid C) > \Pr(E)$, it already follows that $\Pr(\neg C) > 0$. Defining $p \stackrel{\text{def}}{=} \Pr(C)$, the equivalence of the two inequalities follows from the equation $\Pr(E) = p \cdot \Pr(E \mid C) + (1 - p) \cdot \Pr(E \mid \neg C)$, which implies

$$\Pr(E \mid C) - \Pr(E) = (1 - p)(\Pr(E \mid C) - \Pr(E \mid \neg C)).$$

The probability-raising principle alone, however, cannot distinguish between cause and effect as it holds if and only if $\Pr(C \mid E) > \Pr(C \mid \neg E)$ as well. For this reason, additional conditions have to be imposed for causal reasoning. One key condition is temporal priority, which prescribes that a cause has to occur *before* the effect.

This section formalizes both the probability-raising principle as well as the requirement of temporal priority for probabilistic operational models. We draw connections between different ideas from the literature to provide an overview over basic probabilistic notions of causality in the context of formal verification. For this, we assume to have given a DTMC \mathcal{M} with a probability distribution over initial states. This way, the sets Π_φ of paths starting in initial states and fulfilling an LTL property φ are measurable [114] and have a well-defined probability value $\Pr_{\mathcal{M}}(\Pi_\varphi)$, which we also denote by $\Pr_{\mathcal{M}}(\varphi)$. Applying the probability-raising principle and expressing the temporal priority using LTL leads to the following first definition of causality in DTMCs for reachability properties.

► **Definition 1** (reachability-cause). *Let \mathcal{M} be a DTMC with state space S and let $C, E \subseteq S$ be two disjoint sets of states. Then C is a reachability-cause of E if $\Pr_{\mathcal{M}}(\neg EU C) > 0$ and*

$$\Pr_{\mathcal{M}}(\diamond E \mid \neg EU C) > \Pr_{\mathcal{M}}(\diamond E). \quad (3)$$

Note that Equation (3) implies that $\Pr_{\mathcal{M}}(\neg(\neg E \mathcal{U} C)) > 0$. This ensures that also $\Pr_{\mathcal{M}}(\diamond E \mid \neg(\neg E \mathcal{U} C))$ is well-defined and so Equation (3) is equivalent to $\Pr_{\mathcal{M}}(\diamond E \mid \neg E \mathcal{U} C) > \Pr_{\mathcal{M}}(\diamond E \mid \neg(\neg E \mathcal{U} C))$. If there are no paths first reaching the effect E and afterwards the cause C , e.g., because the states in the effect E are absorbing, Equation (3) simplifies to $\Pr_{\mathcal{M}}(\diamond E \mid \diamond C) > \Pr_{\mathcal{M}}(\diamond E)$.

In this treatment of reachability properties, a cause C specifies the set of finite executions ending in C that cause the subsequent extension to an infinite execution to satisfy $\diamond E$. This idea can be lifted to the treatment of causes of arbitrary events in \mathcal{M} specified by a measurable set of infinite paths $\mathcal{L} \subseteq S^\omega$. A cause is then a set of finite paths $\Gamma \subseteq S^+$. Besides the probability-raising property, the temporal priority condition needs to be included. For path properties this needs extra consideration. While for a cause $\Gamma \subseteq S^+$ it is clear that the cause is observed once a finite path in Γ is generated in a DTMC, this is not the case for the effect $\mathcal{L} \subseteq S^\omega$ as it consists of infinite executions. However, it seems natural to say that the effect occurred on a finite path δ whenever $\Pr_{\mathcal{M}}(\mathcal{L} \mid \delta) = 1$, i.e., if a generated finite path ensures that almost all infinite extensions belong to \mathcal{L} . Here, we used δ to also denote the event of all infinite paths having δ as a prefix. Analogously, for a set of finite paths Γ , we denote by Γ also the event of all infinite paths with a prefix in Γ . Consequently, $\neg\Gamma$ denotes the event of all infinite paths that have no prefix in Γ . The discussed treatment of temporal priority is now used in the following definition of a cause in a DTMC.

► **Definition 2** (global PR-cause). *Let \mathcal{M} be a DTMC with state space S , let $\Gamma \subseteq S^+$ be a non-empty set of finite paths, and let $\mathcal{L} \subseteq S^\omega$ be a measurable set of paths. Then, Γ is a global probability-raising cause (global PR-cause) for \mathcal{L} if the following two conditions hold:*

PAC1: $\Pr_{\mathcal{M}}(\mathcal{L} \mid \Gamma) > \Pr_{\mathcal{M}}(\mathcal{L})$, and

PAC2: for all $\gamma \in \Gamma$, no proper prefix γ' of γ satisfies $\Pr_{\mathcal{M}}(\mathcal{L} \mid \gamma') = 1$.

As Γ is a set of finite paths in \mathcal{M} , the cylinder set spanned by each $\gamma' \in \Gamma$ has positive probability. So, the conditional probabilities $\Pr_{\mathcal{M}}(\mathcal{L} \mid \Gamma)$ and $\Pr_{\mathcal{M}}(\mathcal{L} \mid \gamma')$ in Definition 2 are well-defined.

Axiom PAC1 expresses the probability-raising principle. It implies that $\Pr_{\mathcal{M}}(\neg\Gamma) > 0$. As this ensures that all necessary conditional probabilities are well-defined, PAC1 is equivalent to the probability-raising condition $\Pr_{\mathcal{M}}(\mathcal{L} \mid \Gamma) > \Pr_{\mathcal{M}}(\mathcal{L} \mid \neg\Gamma)$. Axiom PAC2 captures that the effect must not occur before the cause.

The requirement on the cause in the provided definition is of *global* nature: the cause Γ as a whole has to guarantee the probability-raising property with respect to the effect. Single elements $\gamma \in \Gamma$, however, do not necessarily guarantee that the probability of the effect has been raised. Furthermore, under mild assumptions, the definition subsumes the treatment of reachability properties above:

► **Proposition 3.** *Let \mathcal{M} be a DTMC with state space S and let $C, E \subseteq S$ be disjoint. Assume that no state in $s \in S \setminus E$ satisfies $\Pr_{\mathcal{M},s}(\diamond E) = 1$. Then the following are equivalent:*

1. C is a reachability-cause for E .
2. The set Γ of finite paths in $(S \setminus (C \cup E))^*C$ is a global PR-cause for the set \mathcal{L} of paths satisfying $\diamond E$.

In the proposition above, $\Pr_{\mathcal{M},s}$ denotes the probability measure induced by \mathcal{M} with assuming s as the unique initial state. A related notion of causality based on probability-raising in DTMCs has been introduced by Kleinberg et al. in a series of papers [75, 76, 74, 66, 122]. Here, probabilistic CTL is used to describe the cause C and the effect E via state formulas. We can describe both events also directly as sets of states in the DTMC by considering exactly those states that fulfil the corresponding probabilistic CTL formula. For reachability

properties, the set C is then said to be a cause of E if $\Pr_{\mathcal{M},c}(\diamond E) > \Pr_{\mathcal{M}}(\diamond E)$ for all $c \in C$. So, the requirement for this notion of causality is *local*: reaching any state $c \in C$ has to ensure that the probability of reaching E afterwards is raised. In case $C = \{c\}$ is a singleton disjoint from E , this notion agrees with Definition 1.

Adapting PAC1 to sets of paths and including the temporal priority requirement that the effect does not occur before the cause (PAC2 as before), we obtain the following definition of causality:

► **Definition 4** (local PR-cause). *Let \mathcal{M} be a DTMC with state space S , $\Gamma \subseteq S^+$ a set of finite paths, and let $\mathcal{L} \subseteq S^\omega$ be a measurable set of paths. Then Γ is a local probability-raising cause (local PR-cause) for \mathcal{L} if*

PAC1^{loc}: *for all $\gamma \in \Gamma$ we have $\Pr_{\mathcal{M}}(\mathcal{L} \mid \gamma) > \Pr_{\mathcal{M}}(\mathcal{L})$, and*

PAC2: *for all $\gamma \in \Gamma$ no proper prefix γ' of γ satisfies $\Pr_{\mathcal{M}}(\mathcal{L} \mid \gamma') = 1$.*

Axiom PAC1^{loc} can be seen as the local version of PAC1. Clearly, PAC1^{loc} implies PAC1. Furthermore, PAC1^{loc} implies that $\Pr_{\mathcal{M}}(\neg\gamma) > 0$ for all $\gamma \in \Gamma$. Hence, we could equivalently reformulate PAC1^{loc} as $\Pr_{\mathcal{M}}(\mathcal{L} \mid \gamma) > \Pr_{\mathcal{M}}(\mathcal{L} \mid \neg\gamma)$ for all $\gamma \in \Gamma$.

The work by Kleinberg et al. proceeds relative to an explicit probability value $p > \Pr_{\mathcal{M}}(\mathcal{L})$ such that $\Pr_{\mathcal{M}}(\mathcal{L} \mid \gamma) \geq p$ for all γ in a cause Γ . The higher this value p lies above $\Pr_{\mathcal{M}}(\mathcal{L})$, the greater is the amount by which all elements of Γ are guaranteed to raise the probability of the effect \mathcal{L} .

Such a reference to a specific threshold value p has also been incorporated into a notion of *p-causes* in [9]. Motivated by monitoring applications (see, e.g., [87]), the underlying idea is that notions of causality could be used to foresee undesirable behavior. If a cause for an erroneous execution is observed, countermeasures can be taken before the error actually occurs. Here it is particularly useful to specify a sensitivity p that expresses how likely an error is after observing the cause. In addition, the occurrence of an erroneous execution should not stay undetected. Therefore, an additional condition is imposed on *p-causes*: almost all executions that exhibit the error should have a prefix in the cause. Together with the temporal priority of the cause (PAC2) as before, these requirements lead to the following definition:

► **Definition 5** (*p-cause*). *Let \mathcal{M} be a DTMC with state space S and $p \in (0, 1]$. A non-empty set $\Gamma \subseteq S^+$ is a *p-cause* for a measurable set $\mathcal{L} \subseteq S^\omega$ if*

PAC1^P: *for all $\gamma \in \Gamma$ we have $\Pr_{\mathcal{M},s_0}(\mathcal{L} \mid \gamma) \geq p$,*

PAC2: *for all $\gamma \in \Gamma$ no proper prefix γ' of γ satisfies $\Pr_{\mathcal{M}}(\mathcal{L} \mid \gamma') = 1$, and*

PAC3: $\Pr_{\mathcal{M}}(\Gamma \mid \mathcal{L}) = 1$.

Besides the practicality in monitoring applications, condition PAC3 also adds the counterfactual idea to the definition. Since almost all executions in \mathcal{L} have a prefix in Γ , the effect occurs with probability 0 if the cause was not observed. Condition PAC1^P is a third variant of the probability-raising requirement that compares the probability of the effect after observing an element of the cause to a specific threshold value p rather than the overall probability $\Pr_{\mathcal{M}}(\mathcal{L})$. In case $p > \Pr_{\mathcal{M}}(\mathcal{L})$, this variant implies PAC1 and PAC1^{loc}.

For ω -regular \mathcal{L} there always exist *p-causes* for any $p \in (0, 1]$. The reason is that then almost all paths in \mathcal{L} have a prefix π with $\Pr_{\mathcal{M}}(\mathcal{L} \mid \pi) = 1$. Choosing the shortest of such prefixes in accordance with condition PAC2 yields a 1-cause and hence a *p-cause* for any p . For $p < 1$, there are multiple *p-causes* in general. In [9], the problem to find cost-optimal *p-causes* with respect to a variety of cost measures is addressed.

The relationship between the different notions of causes is summarized in the following proposition. It is a direct consequence of the implications between the axioms used in the definitions that have been discussed so far.

► **Proposition 6.** *Let \mathcal{M} be a DTMC with state space S , $\Gamma \subseteq S^+$, and let $\mathcal{L} \subseteq S^\omega$ be a measurable set of paths. Then the following statements hold:*

1. *If Γ is a p -cause for \mathcal{L} for some $p > \Pr_{\mathcal{M}}(\mathcal{L})$, then Γ is also a local PR-cause for \mathcal{L} .*
2. *If Γ is a local PR-cause for \mathcal{L} , then Γ is also a global PR-cause for \mathcal{L} .*
3. *If Γ is a singleton, then Γ is a local PR-cause for \mathcal{L} iff Γ is a global PR-cause for \mathcal{L} .*

The probabilistic notions of causality discussed in this section naturally constitute forward-looking notions: the probability-raising principle inherently addresses the behavior of a system across multiple executions, and causes are prone to exhibiting a predictive character. These notions can be useful in inferring causal dependencies in data series [75, 74, 66] and predicting undesirable behavior of reactive systems through runtime monitoring [9]. Nevertheless, as far as formal probabilistic models are concerned, a comprehensive study of cause-effect relationships is still at the beginning.

5 Concluding Remarks

This article gave an overview of recent trends in causality-based reasoning in the verification context. The focus of this article was on concepts that aim to explicate *why* a system exhibits a specific observable behavior and to which degree individual agents of a system can be held *responsible* for it. For non-probabilistic formal models, concepts of causation have been introduced in multiple facets and examined for manifold applications. To increase the power of causal inferences, a more systematic study relating forward and backward notions of causality would be highly beneficial.

Compared to the non-probabilistic setting, research on probabilistic causation in stochastic operational models is still in its infancy. While the techniques presented here are limited to purely probabilistic models (Markov chains), an examination of causality in probabilistic models with nondeterminism (Markov decision processes) is largely open. A first step in this direction is a formalization of action causes as a hyperproperty in Markov decision processes [37]. Another important future direction is to reason about cause-effect relationships in hidden Markovian models where states (and events) are not fully observable.

Another research strand not covered in this article are causality-based verification techniques (see, e.g., [82, 83]) that rely on the successive identification of cause-effect relationships between events to generate a causality-based proof for the satisfaction or violation of a system property. Along these lines, the work [8] presents a causality-based technique for solving symbolically expressed, infinite-state two-player reachability games. Applying this paradigm also in a probabilistic setting is a promising direction of study.

References

- 1 Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values, 2020. [arXiv:1903.10464](https://arxiv.org/abs/1903.10464).
- 2 Erika Ábrahám and Borzoo Bonakdarpour. HyperPCTL: A temporal logic for probabilistic hyperproperties. In *Proc. of the 15th Intern. Conf. on Quantitative Evaluation of Systems (QEST)*, pages 20–35. Springer, 2018.
- 3 R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995. Hybrid Systems. doi:10.1016/0304-3975(94)00202-T.

- 4 Rajeev Alur and David L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126(2):183–235, 1994. doi:10.1016/0304-3975(94)90010-8.
- 5 Sven Apel and Christian Kästner. An Overview of Feature-Oriented Software Development. *Journal of Object Technology*, 8:49–84, 2009.
- 6 Uwe Aßmann, Christel Baier, Clemens Dubslaff, Dominik Grzelak, Simon Hanisch, Ardhi P. P. Hartono, Stefan Köpsell, Tianfang Lin, and Thorsten Strufe. *Tactile computing: Essential building blocks for the Tactile Internet*, chapter 13, pages 301–326. Academic Press, 2021.
- 7 R. J. Aumann and L. S. Shapley. *Values of Non-Atomic Games*. Princeton University Press, 1974. URL: <http://www.jstor.org/stable/j.ctt13x149m>.
- 8 Christel Baier, Norine Coenen, Bernd Finkbeiner, Florian Funke, Simon Jantsch, and Julian Siber. Causality-based Game Solving. In *Proc. of the 33rd Intern. Conf. on Computer Aided Verification (CAV)* (to appear), 2021.
- 9 Christel Baier, Florian Funke, Simon Jantsch, Jakob Piribauer, and Robin Ziemek. Probabilistic causes in Markov chains, 2021. arXiv:2104.13604.
- 10 Christel Baier, Florian Funke, and Rupak Majumdar. A Game-Theoretic Account of Responsibility Allocation. In *Proc. of the 30th International Joint Conference on Artificial Intelligence (IJCAI)* (to appear), 2021.
- 11 Christel Baier, Florian Funke, and Rupak Majumdar. Responsibility Attribution in Parameterized Markovian Models. In *Proc. of the 35th AAAI Conference on Artificial Intelligence (AAAI)* (to appear), 2021.
- 12 Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.
- 13 Thomas Ball, Mayur Naik, and Sriram K. Rajamani. From symptom to cause: Localizing errors in counterexample traces. *SIGPLAN Not.*, 38(1):97–105, 2003. doi:10.1145/640128.604140.
- 14 Sander Beckers and Joost Vennekens. A General Framework for Defining and Extending Actual Causation Using CP-Logic. *Int. J. Approx. Reasoning*, 77(C):105–126, October 2016. doi:10.1016/j.ijar.2016.05.008.
- 15 Adrian Beer, Stephan Heidinger, Uwe Kühne, Florian Leitner-Fischer, and Stefan Leue. Symbolic Causality Checking Using Bounded Model Checking. In *Model Checking Software*, pages 203–221. Springer, 2015.
- 16 Ilan Beer, Shoham Ben-David, Hana Chockler, Avigail Orni, and Richard Trefler. Explaining counterexamples using causality. In *Proc. of the 21st Intern. Conf. on Computer Aided Verification (CAV)*, pages 94–108. Springer, 2009. doi:10.1007/978-3-642-02658-4_11.
- 17 Ilan Beer, Shoham Ben-David, Cindy Eisner, and Yoav Rodeh. Efficient Detection of Vacuity in ACTL Formulas. In *Proc. of the 9th Intern. Conf. on Computer Aided Verification (CAV)*, pages 279–290, 1997. doi:10.1007/3-540-63166-6_28.
- 18 Matthew Braham and Martin van Hees. An Anatomy of Moral Responsibility. *Mind*, 121(483):601–634, 2012.
- 19 Jan Broersen. Deontic epistemic stit logic distinguishing modes of mens rea. *Journal of Applied Logic*, 9(2):137–152, 2011. doi:10.1016/j.jal.2010.06.002.
- 20 Georgiana Caltais, Sophie Linnea Guetlein, and Stefan Leue. Causality for General LTL-definable Properties. In *Proc. of the 3rd Workshop on formal reasoning about Causation, Responsibility, and Explanations in Science and Technology (CREST)*, pages 1–15, 2018. doi:10.4204/EPTCS.286.1.
- 21 Nancy Cartwright. Causal Laws and Effective Strategies. *Noûs*, 13(4):419–437, 1979. doi:10.1093/0198247044.003.0002.
- 22 Krishnendu Chatterjee and Thomas A. Henzinger. A Survey of Stochastic ω -Regular Games. *J. Comput. Syst. Sci.*, 78(2):394–413, 2012. doi:10.1016/j.jcss.2011.05.002.
- 23 Hana Chockler and Joseph Y. Halpern. Responsibility and Blame: A Structural-Model Approach. *J. Artif. Int. Res.*, 22(1):93–115, October 2004.
- 24 Hana Chockler, Joseph Y. Halpern, and Orna Kupferman. What causes a system to satisfy a specification? *ACM Transactions on Computational Logic*, 9(3):20:1–20:26, 2008. doi:10.1145/1352582.1352588.

- 25 Hana Chockler, Orna Kupferman, Robert P. Kurshan, and Moshe Y. Vardi. A Practical Approach to Coverage in Model Checking. In *Proc. of the 13th Intern. Conf. on Computer Aided Verification (CAV)*, pages 66–78, 2001. doi:10.1007/3-540-44585-4_7.
- 26 Hana Chockler, Orna Kupferman, and Moshe Vardi. Coverage Metrics for Formal Verification. *Intern. Journal on Software Tools for Technology Transfer (STTT)*, 8(4–5):373–386, 2006.
- 27 Hana Chockler, Orna Kupferman, and Moshe Y. Vardi. Coverage Metrics for Temporal Logic Model Checking. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 528–542, 2001. doi:10.1007/3-540-45319-9_36.
- 28 Philipp Chrszon, Clemens Dubslaff, Sascha Klüppelholz, and Christel Baier. ProFeat: feature-oriented engineering for family-based probabilistic model checking. *Formal Aspects of Computing*, 30(1):45–75, 2018.
- 29 Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8:244–263, 1986.
- 30 Edmund M. Clarke, Orna Grumberg, Kenneth L. McMillan, and Xudong Zhao. Efficient Generation of Counterexamples and Witnesses in Symbolic Model Checking. In *Proc. of the 32nd Annual ACM/IEEE Design Automation Conf. (DAC)*, pages 427–432, New York, NY, USA, 1995. ACM. doi:10.1145/217474.217565.
- 31 Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem. *Handbook of Model Checking*. Springer Publishing Company, Incorporated, 1st edition, 2018.
- 32 Andreas Classen, Patrick Heymans, Pierre-Yves Schobbens, Axel Legay, and Jean-François Raskin. Model Checking Lots of Systems: Efficient Verification of Temporal Properties in Software Product Lines. In *Proc. of the 32nd Intern. Conf. on Software Engineering (ICSE)*, pages 335–344. ACM, 2010.
- 33 Norine Coenen. Causality and Hyperproperties. In Gregor Gössler, Stefan Leue, and Shin Nakajima, editors, *Causal Reasoning in Systems (NII Shonan Meeting 139)*, 2019. URL: <https://shonan.nii.ac.jp/seminars/139/>.
- 34 Anupam Datta, Deepak Garg, Dilsun Kaynar, Divya Sharma, and Arunesh Sinha. Program Actions as Actual Causes: A Building Block for Accountability. In *Proc. of the 28th IEEE Computer Security Foundations Symp. (CSF)*, pages 261–275, 2015. doi:10.1109/CSF.2015.25.
- 35 Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *Proc. of the 37th IEEE Symp. on Security and Privacy (SP)*, pages 598–617, 2016. doi:10.1109/SP.2016.42.
- 36 David L. Dill. Timing Assumptions and Verification of Finite-State Concurrent Systems. In *Automatic Verification Methods for Finite State Systems*, LNCS. Springer, 1990. doi:10.1007/3-540-52148-8_17.
- 37 Rayna Dimitrova, Bernd Finkbeiner, and Hazem Torfah. Probabilistic Hyperproperties of Markov Decision Processes. In *Proc. of the 18th Intern. Symp. on Automated Technology for Verification and Analysis (ATVA)*, volume 12302 of LNCS, pages 484–500. Springer, 2020.
- 38 Clemens Dubslaff, Christel Baier, and Sascha Klüppelholz. Probabilistic model checking for feature-oriented systems. *Trans. Aspect-Oriented Software Development*, 12:180–220, 2015.
- 39 Ellery Eells. *Probabilistic Causality*. Cambridge Studies in Probability, Induction and Decision Theory. Cambridge University Press, 1991.
- 40 Thomas Eiter and Thomas Lukasiewicz. Complexity results for explanations in the structural-model approach. *Artificial Intelligence*, 154(1-2):145–198, 2004. doi:10.1016/j.artint.2003.06.002.
- 41 Joel Feinberg. *Doing & Deserving: Essays in the Theory of Responsibility*. Princeton University Press, Princeton, USA, 1970.
- 42 Luke Fenton-Glynn. A Proposed Probabilistic Extension of the Halpern and Pearl Definition of ‘Actual Cause’. *British Journal for the Philosophy of Science*, 68(4):1061–1124, 2017. doi:10.1093/bjps/axv056.

- 43 Eric Friedman and Hervé Moulin. Three methods to share joint costs or surplus. *Journal of Economic Theory*, 87(2):275–312, 1999.
- 44 Florian Funke, Simon Jantsch, and Christel Baier. Farkas Certificates and Minimal Witnesses for Probabilistic Reachability Constraints. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Springer, 2020. doi:10.1007/978-3-030-45190-5_18.
- 45 David Galles and Judea Pearl. Axioms of Causal Relevance. *Artif. Intell.*, 97(1–2):9–43, December 1997. doi:10.1016/S0004-3702(97)00047-7.
- 46 David Galles and Judea Pearl. An Axiomatic Characterization of Causal Counterfactuals. *Foundations of Science*, 3:151–182, 1998. doi:10.1023/A:1009602825894.
- 47 Edmund Gettier. Is justified true belief knowledge? *Analysis*, 23:121–123, 1963.
- 48 Robert Givan, Sonia Leach, and Thomas Dean. Bounded-parameter Markov decision processes. *Artificial Intelligence*, 122(1):71–109, 2000. doi:10.1016/S0004-3702(00)00047-3.
- 49 Gregor Gössler and Daniel Le Métayer. A General Trace-Based Framework of Logical Causality. In *Formal Aspects of Component Software*, pages 157–173. Springer, 2014.
- 50 Gregor Gössler and Daniel Le Métayer. A General Framework for Blaming in Component-Based Systems. *Science of Computer Programming*, 113:223–235, 2015. doi:10.1016/j.scico.2015.06.010.
- 51 Gregor Gössler and Jean-Bernard Stefani. Causality Analysis and Fault Ascription in Component-Based Systems. *Theoretical Computer Science*, 837:158–180, 2020. doi:10.1016/j.tcs.2020.06.010.
- 52 Stephen R. Grimm. Understanding as Knowledge of Causes. In Abrol Fairweather, editor, *Virtue Epistemology Naturalized: Bridges Between Virtue Epistemology and Philosophy of Science*, pages 329–345. Springer, 2014. doi:10.1007/978-3-319-04672-3_19.
- 53 Alex Groce. Error Explanation with Distance Metrics. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 108–122, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- 54 Ned Hall. Structural Equations and Causation. *Philosophical Studies*, 132(1):109–136, 2007. URL: <http://www.jstor.org/stable/25471849>.
- 55 Joseph Y. Halpern. Axiomatizing Causal Reasoning. *Journal of Artif. Intell. Research*, 12(1):317–337, 2000.
- 56 Joseph Y. Halpern. A Modification of the Halpern-Pearl Definition of Causality. In *Proc. of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3022–3033. AAAI Press, 2015.
- 57 Joseph Y. Halpern. *Actual Causality*. MIT Press, 2016.
- 58 Joseph Y. Halpern and Judea Pearl. Causes and Explanations: A Structural-Model Approach: Part i: Causes. In *Proc. of the 17th Conf. on Uncertainty in AI (UAI)*, pages 194–202. Morgan Kaufmann Publishers Inc., 2001.
- 59 Joseph Y. Halpern and Judea Pearl. Causes and Explanations: A Structural-Model Approach. Part I: Causes. *The British Journal for the Philosophy of Science*, 56(4):843–887, 2005.
- 60 Joseph Y. Halpern and Judea Pearl. Causes and Explanations: A Structural-Model Approach. Part II: Explanations. *The British Journal for the Philosophy of Science*, 56(4):889–911, 2005.
- 61 Mark Harman and Robert Hierons. An overview of program slicing. *Software Focus*, 2(3):85–92, 2001. doi:10.1002/swf.41.
- 62 Thomas A. Henzinger, Ranjit Jhala, Rupak Majumdar, George C. Necula, Grégoire Sutre, and Westley Weimer. Temporal-Safety Proofs for Systems Code. In *Proc. of the 14th Intern. Conf. on Computer Aided Verification (CAV)*, volume 2404 of *LNCS*, pages 526–538. Springer, 2002. doi:10.1007/3-540-45657-0_45.
- 63 Christopher Hitchcock. The Intransitivity of Causation Revealed in Equations and Graphs. *The Journal of Philosophy*, 98(6):273–299, 2001. URL: <http://www.jstor.org/stable/2678432>.
- 64 Martin Hofmann, Christian Neukirchen, and Harald Rueß. Certification for μ -Calculus with Winning Strategies. In *Proc. of the 23rd Intern. Symp. on Model Checking Software (SPIN)*, volume 9641 of *LNCS*, pages 111–128. Springer, 2016. doi:10.1007/978-3-319-32582-8_8.

- 65 Yatin Hoskote, Timothy Kam, Pei-Hsin Ho, and Xudong Zhao. Coverage Estimation for Symbolic Model Checking. In *Proc. of the 36th Annual ACM/IEEE Design Automation Conf. (DAC)*, pages 300–305, 1999. doi:10.1145/309847.309936.
- 66 Yuxiao Huang and Samantha Kleinberg. Fast and Accurate Causal Inference from Time Series Data. In *Proc. of the 28th Intern. Florida AI Research Society Conf. (FLAIRS)*, pages 49–54. AAAI Press, 2015. URL: <http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS15/paper/view/10434>.
- 67 David Hume. *A Treatise of Human Nature*. John Noon, 1739.
- 68 David Hume. *An Enquiry Concerning Human Understanding*. London, 1748.
- 69 Amjad Ibrahim and Alexander Pretschner. From checking to inference: Actual causality computations as optimization problems. In *Proc. of the 18th Intern. Symp. on Automated Technology for Verification and Analysis (ATVA)*, pages 343–359. Springer, 2020. doi:10.1007/978-3-030-59152-6_19.
- 70 Simon Jantsch, Florian Funke, and Christel Baier. Minimal Witnesses for Probabilistic Timed Automata. In *Proc. of the 18th Intern. Symp. on Automated Technology for Verification and Analysis (ATVA)*, pages 501–517. Springer, 2020. doi:10.1007/978-3-030-59152-6_28.
- 71 Bengt Jonsson and Kim G. Larsen. Specification and refinement of probabilistic processes. In *Proc. of the 6th Annual IEEE Symp. on Logic in Computer Science (LICS)*, pages 266–277, 1991.
- 72 Manu Jose and Rupak Majumdar. Cause clue clauses: error localization using maximum satisfiability. In *Proc. of the 32nd ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI)*, pages 437–446. ACM, 2011. doi:10.1145/1993498.1993550.
- 73 Severin Kacianka, Amjad Ibrahim, and Alexander Pretschner. Expressing Accountability Patterns using Structural Causal Models, 2020. arXiv:2005.03294.
- 74 Samantha Kleinberg. A Logic for Causal Inference in Time Series with Discrete and Continuous Variables. In *Proc. of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 943–950, 2011. doi:10.5591/978-1-57735-516-8/IJCAI11-163.
- 75 Samantha Kleinberg and Bud Mishra. The Temporal Logic of Causal Structures. In *Proc. of the 25th Conf. on Uncertainty in AI (UAI)*, pages 303–312, 2009.
- 76 Samantha Kleinberg and Bud Mishra. The Temporal Logic of Token Causes. In *Proc. of the 12th Intern. Conf. on Principles of Knowledge Representation and Reasoning (KR)*, 2010.
- 77 Martin Kölbl and Stefan Leue. An Efficient Algorithm for Computing Causal Trace Sets in Causality Checking. In *Proc. of the 17th Intern. Symp. on Automated Technology for Verification and Analysis (ATVA)*, pages 171–186. Springer, 2019. doi:10.1007/978-3-030-31784-3_10.
- 78 Martin Kölbl, Stefan Leue, and Robert Schmid. Dynamic Causes for the Violation of Timed Reachability Properties. In *Proc. of the 18th Intern. Conf. on Formal Modeling and Analysis of Timed Systems (FORMATS)*, pages 127–143. Springer, 2020. doi:10.1007/978-3-030-57628-8_8.
- 79 Igor Kozine and Lev Utkin. Interval-Valued Finite Markov Chains. *Reliable Computing*, 8:97–113, April 2002. doi:10.1023/A:1014745904458.
- 80 Harold W. Kuhn. *Extensive Games and the Problem of Information*, pages 193–216. Princeton University Press, Princeton, 1953. doi:10.1515/9781400881970-012.
- 81 Orna Kupferman and Moshe Y. Vardi. Vacuity Detection in Temporal Model Checking. In *Proc. of the 10th IFIP Working Conf. on Correct Hardware Design and Verification Methods (CHARME)*, pages 82–96, 1999.
- 82 Andrey Kupriyanov and Bernd Finkbeiner. Causality-Based Verification of Multi-threaded Programs. In *Proc. of the 24th Intern. Conf. on Concurrency Theory (CONCUR)*, LNCS, pages 257–272. Springer, 2013. doi:10.1007/978-3-642-40184-8_19.
- 83 Andrey Kupriyanov and Bernd Finkbeiner. Causal Termination of Multi-threaded Programs. In *Proc. of the 26th Intern. Conf. on Computer Aided Verification (CAV)*, LNCS, pages 814–830. Springer, 2014. doi:10.1007/978-3-319-08867-9_54.

- 84 Igal Kwart. Causation: Probabilistic and Counterfactual Analyses. In *Causation and Counterfactuals*, pages 359–387. MIT Press, Cambridge, MA, USA, 2004.
- 85 Kim G. Larsen and Bent Thomsen. A Modal Process Logic. In *Proc. of the 3rd Annual Symp. on Logic in Computer Science (LICS)*, pages 203–210, 1988. doi:10.1109/LICS.1988.51119.
- 86 Florian Leitner-Fischer and Stefan Leue. Causality Checking for Complex System Models. In *Proc. of the 14th Intern. Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, pages 248–267, 2013. doi:10.1007/978-3-642-35873-9_16.
- 87 Martin Leucker and Christian Schallhart. A brief account of runtime verification. *The Journal of Logic and Algebraic Programming*, 78(5):293–303, 2009. doi:10.1016/j.jlap.2008.08.004.
- 88 David Lewis. Causation. *Journal of Philosophy*, 70(17):556–567, 1973. doi:10.2307/2025310.
- 89 David Lewis. Postscripts to ‘Causation’. In *Philosophical Papers Vol. II*. Oxford University Press, 1986.
- 90 Harry R. Lewis. A logic of concrete time intervals. In *Proc. of the 5th Annual IEEE Symp. on Logic in Computer Science (LICS)*, pages 380–389, 1990. doi:10.1109/LICS.1990.113763.
- 91 Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Proc. of the 31st Intern. Conf. on Neural Information Processing Systems (NeurIPS)*, pages 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.
- 92 Donald A. Martin. The Determinacy of Blackwell Games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998. URL: <http://www.jstor.org/stable/2586667>.
- 93 Corto Mascle, Christel Baier, Florian Funke, Simon Jantsch, and Stefan Kiefer. Responsibility and verification: Importance value in temporal logics. In *Proc. of the 36th Annual Symp. on Logic in Computer Science (LICS)* (to appear), 2021.
- 94 Pavel Naumov and Jia Tao. An epistemic logic of blameworthiness. *Artificial Intelligence*, 283:103269, 2020. doi:10.1016/j.artint.2020.103269.
- 95 Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995. URL: <http://www.jstor.org/stable/2337329>.
- 96 Judea Pearl. On the Definition of Actual Cause. *Technical Report R-259, Computer Science Dept., UCLA*, 1998. doi:10.1.1.53.9540.
- 97 Judea Pearl. *Causality*. Cambridge University Press, 2 edition, 2009. doi:10.1017/CB09780511803161.
- 98 Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.
- 99 Amir Pnueli. The Temporal Logic of Programs. In *Proc. of the 18th Annual Symp. on Foundations of Computer Science (FOCS)*, pages 46–57, 1977. doi:10.1109/SFCS.1977.32.
- 100 Duncan Pritchard. Knowledge and understanding. In *Virtue Epistemology Naturalized: Bridges between Virtue Epistemology and Philosophy of Science*, Synthese Library 366, pages 315–327. Springer, 2014.
- 101 Duncan Pritchard. *Epistemology*. Palgrave Macmillan, 2016.
- 102 Mitra Purandare and Fabio Somenzi. Vacuum cleaning CTL formulae. In *Proc. of the 14th Intern. Conf. on Computer Aided Verification (CAV)*, pages 485–499, 2002. doi:10.1007/3-540-45657-0_39.
- 103 Hans Reichenbach. Die Kausalstruktur der Welt und der Unterschied von Vergangenheit und Zukunft. *Sitzungsberichte der Bayerische Akademie der Wissenschaft*, 2:81–119, 1925.
- 104 Hans Reichenbach. *The Direction of Time*. Berkeley and Los Angeles: University of California Press, 1956.
- 105 Manos Renieres and Steven P. Reiss. Fault localization with nearest neighbor queries. In *Proc. of the 18th IEEE Intern. Conf. on Automated Software Engineering (ASE)*, pages 30–39, 2003. doi:10.1109/ASE.2003.1240292.
- 106 Lloyd S. Shapley. A value for n -person games. In *Contributions to the Theory of Games. Vol. II*, pages 307–317. Princeton University Press, 1953.
- 107 Brian Skyrms. Causal Necessity. *Philosophy of Science*, 48(2):329–335, 1981. doi:10.1086/289003.

- 108 Jeremy Sproston. Decidable Model Checking of Probabilistic Hybrid Automata. In Mathai Joseph, editor, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 31–45, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
- 109 Yi Sun and Mukund Sundararajan. Axiomatic attribution for multilinear functions. *Proc. of the 12th ACM Conf. on Electronic Commerce (EC)*, 2011. doi:10.1145/1993574.1993601.
- 110 Mukund Sundararajan and Amir Najmi. The Many Shapley Values for Model Explanation. In *Proc. of the 37th Intern. Conf. on Machine Learning (ICML)*, volume 119 of *Machine Learning Research*, pages 9269–9278. PMLR, 2020.
- 111 Patrick Suppes. *A Probabilistic Theory of Causality*. Amsterdam: North-Holland Pub. Co., 1970.
- 112 Thomas Thüm, Sven Apel, Christian Kästner, Ina Schaefer, and Gunter Saake. A Classification and Survey of Analysis Strategies for Software Product Lines. *ACM Comput. Surv.*, 47(1s):6:1–6:45, 2014.
- 113 Ibo van de Poel. The Relation Between Forward-Looking and Backward-Looking Responsibility. In *Moral Responsibility: Beyond Free Will and Determinism*, pages 37–52, Dordrecht, 2011. Springer. doi:10.1007/978-94-007-1878-4_3.
- 114 Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. of the 26th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 327–338. IEEE Computer Society, 1985.
- 115 Joost Vennekens, Marc Denecker, and Maurice Bruynooghe. CP-logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming*, 9(3):245–308, 2009. doi:10.1017/S1471068409003767.
- 116 John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.
- 117 Erik Štrumbelj and Igor Kononenko. Explaining Prediction Models and Individual Predictions with Feature Contributions. *Knowl. Inf. Syst.*, 41(3):647–665, 2014. doi:10.1007/s10115-013-0679-x.
- 118 Chao Wang, Zijiang Yang, Franjo Ivancic, and Aarti Gupta. Whodunit? Causal Analysis for Counterexamples. In *Proc. of the 4th Intern. Symp. on Automated Technology for Verification and Analysis (ATVA)*, pages 82–95, 2006. doi:10.1007/11901914_9.
- 119 Vahid Yazdanpanah, Mehdi Dastani, Wojciech Jamroga, Natasha Alechina, and Brian Logan. Strategic Responsibility Under Imperfect Information. In *Proc. of the 18th Intern. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 592–600. AAMAS Foundation, 2019.
- 120 Andreas Zeller. Isolating Cause-Effect Chains from Computer Programs. In *Proc. of the 10th ACM SIGSOFT Symp. on Foundations of Software Engineering (FSE)*, pages 1–10, New York, NY, USA, 2002. ACM. doi:10.1145/587051.587053.
- 121 Danfeng Zhang, Andrew C. Myers, Dimitrios Vytiniotis, and Simon L. Peyton Jones. SHerrLoc: A Static Holistic Error Locator. *ACM Trans. Program. Lang. Syst.*, 39(4):18:1–18:47, 2017. doi:10.1145/3121137.
- 122 Min Zheng and Samantha Kleinberg. A method for automating token causal explanation and discovery. In *Proc. of the 13th Florida AI Research Society Conf. (FLAIRS)*, 2017.