Causal, Strategic, and Combined Responsibility Anticipation and Attribution in Situation Calculus Concurrent Game Structures

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

University of Regina

By

MohammadHossein Karimian

Regina, Saskatchewan

December 2025

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

MohammadHossein Karimian, candidate for the degree of **Master of Science in Computer Science,** has presented a thesis titled, ***Causal, strategic, and combined responsibility anticipation and attribution in situation calculus concurrent game structures,*** in an oral examination held on **December 10, 2025**. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:	Dr. Paul Simard Smith, Department of Philosophy and Classics
Supervisor(s):	Dr. Shakil M. Khan, Department of Computer Science
Committee Member:	Dr. Malek Mouhoub, Department of Computer Science
Chair of Defense:	Dr. Mohammad Khondoker, Industrial Systems Engineering

Abstract

Responsibility is at the heart of accountable multi-agent decision-making. With increasing autonomy and complexity in AI, there is a need for frameworks that connect reasoning with explanations grounded in what actually produced an outcome and who caused it (causal responsibility), as well as what could have prevented it (strategic responsibility).

I develop a unifying treatment of responsibility within a synchronous game-theoretic setting based on situation calculus synchronous game structures (SCSGS), where multiple agents may act concurrently. My first contribution is an account of actual causation based on joint moves, where a strict minimality condition prevents over-determination while being immune to preemption. This proposal resolves long-standing issues by capturing actual causation in more realistic concurrent domains. On top of this causal layer, I formalize various responsibility notions. The active and passive strategic notions evaluate whether an agent ensured or could have prevented the outcome with available strategies, respectively; the causal notion, which is novel to this thesis, states whether the agent's contribution was necessary according to some minimal causal chain. I show that these notions are extensionally distinct

and introduce a combined attribution that considers both strategic intent and causal contribution. I study some interesting properties of responsibility, including their persistence. I then consider a formal example to analyze group responsibility.

Keywords: Actual Causality; Strategic responsibility; Causal Responsibility; Reasoning about actions; Situation Calculus Synchronous Game Structures (SCSGS); Concurrent Multi-Agent Systems; Logic; Knowledge Representation;

Acknowledgments

I would like to begin my thesis by thanking my supervisor, Dr. Shakil M. Khan, and expressing my sincere gratitude to him for his kind support during my thesis, as well as for his guidance and assistance in completing my thesis. Also, I am deeply grateful to Prof. Yves Lespérance for his continuous help and support during my research. I would also like to acknowledge Dr. Paul Simard Smith for serving as the external examiner at my defense and Dr. Malek Mouhoub for his service as a committee member. I am sincerely grateful to Dr. Lisa Fan for her insightful comments on my thesis. Additionally, I like to thank the Faculty of Graduate Studies and Research at the University of Regina for funding my research, and the Natural Sciences and Engineering Research Council of Canada for its support through grants awarded to my outstanding supervisor. I am forever indebted to my family for their love, support, and I thank them from the depths of my heart. Finally, I want to send my regards to the researchers who built the bases of this thesis; without their work, this research would have not been possible.

Contents

List of Figures

Transparency Statement

In this thesis, I used **Grammarly** to correct grammatical errors and used **ChatGPT** only to paraphrase sentences.

I have my supervisor's approval for this limited use of these technologies. No other generative AI systems were used in writing this thesis or any of my related papers.

I recognize that AI tools can produce biased or incomplete text, and I have ensured that there are no inaccurate claims. I take full responsibility for the accuracy, originality, and academic integrity of this work.

Statement of Contributions

This manuscript-style thesis develops a unified account of *actual causation* and *responsibility* in situation calculus concurrent game structures (SCSGS). The thesis proceeds in two main parts. Chapter 4 (*On the Semantics of Actual Causality in Situation Calculus Concurrent Game Structures*, 4) defines actual causation for synchronous multi-agent domains, identifies the joint moves that bring about the relevant effect, and organizes them into causal chains that may include multiple minimal sufficient sets. Chapter 5 (*Causal Responsibility Anticipation and Attribution in Situation Calculus Concurrent Game Structures*, 5) then builds upon this causal semantics to formalize strategic, passive (ex-ante and ex-post), causal, and combined notions of responsibility for both individual agents and coalitions.

Author's contributions. I developed the formal definitions, proofs, and examples, and wrote the first draft of both papers under the guidance of my supervisor (Dr. Shakil M. Khan) and co-author (Prof. Yves Lespérance). Dr. Shakil M. Khan and Prof. Yves Lespérance also provided conceptual feedback, technical suggestions, and editorial comments.

Copyright and permissions. Prior work is cited and documented in the references. The two papers are reproduced in Chapter 4 and Chapter 5 with minor formatting and figure adjustments to align with the thesis.

Chapter 1

Introduction

1.1 Background

As modern AI systems grow in autonomy and complexity, there is a growing need for them to address issues in AI ethics. Responsibility is a central idea behind multi-agent, accountable decision-making. Causation and responsibility are closely related concepts. *Actual causality* asks what brought about an observed outcome in a given scenario and involves identifying the precise actions/events that caused the observed effect, given the history of actions/events (i.e., the scenario). It is different from *General Causality*, which studies the general causal relationship between events. For example, it is perceived that generally smoking is a cause of lung cancer, and lung cancer is, in general, a cause of death. This is very different from studying whether a particular accident was the cause of John's paralysis, which is actual causality. Work on responsibility involves determining who should be held accountable for a given outcome, either due to something they did or for not behaving in certain ways.

1.2 Motivation and Specific Problem

Pearl [1] was the first to advance a computational framework for actual causation. Based on Pearl's work, Halpern and Pearl [2, 3] and others [4] later developed this further. These proposals, which are founded on structural equation models, were criticized for having expressive limitations [5]. These also fail to handle some examples. To deal with this, researchers have recently worked on formalizing actual causes in formal action-theoretic frameworks, in particular in the situation calculus [6]. However, a major limitation of these proposals is that they take the scenario to be a sequence of actions. In other words, while these accommodate actions by multiple agents, these actions can only occur in a turn-taking fashion, and concurrent actions by different agents are not allowed.

Independently, there has been much work on responsibility to address accountability in AI, including strategic responsibility [7]. However, most of these are limited to single agents; also, the connection between actual causality and responsibility remains largely unexplored. In particular, to the best of my knowledge, no computational study of responsibility due to actual causal contribution has yet been developed.

1.3 Contributions

In a direct response to the challenges presented above, in this thesis, I define a formal model of actual causation within situation calculus concurrent game structures, a game theoretic logic framework that allows concurrent moves by multiple agents.

Based on this, I propose two novel notions of responsibility by accommodating a causal perspective. The contributions of this thesis are outlined below.

1. Accommodation of concurrent scenarios:

   My proposal extends previous approaches to actual causation in the situation calculus by modeling scenarios with concurrent actions. In my framework, at every step, each agent synchronously and concurrently makes a move, and the situation/state is updated based on their joint moves. In this framework, I first show how one can identify which joint moves caused the observed effect. I organize these causes in a *causal chain*. As we will see, in a synchronous concurrent game theoretic setting, it is possible for such a causal chain to include multiple independent sets of causes, each of which is sufficient to bring about the effect. I show how one can untangle the causal chain and extract these sets.

2. Systematic Analysis of Causal and Strategic Responsibility:

   Secondly, I extend previous proposals of strategic responsibility that were defined for single agents to allow for coalitions of agents. In particular, I define both active and passive notions of responsibility. The former pertains to an agent ensuring some state of affairs occurs through their actions, while the latter involves the agent's failure to prevent that effect from occurring [7]. Following [8], I also define two variants of passive responsibility, based on whether the reasoning occurs before or after the outcome. Passive responsibility anticipation

is a future-looking or ex-ante notion and involves determining whether a certain choice would incur some responsibility. Passive responsibility attribution, on the other hand, is a retrospective or ex-post notion, which involves assigning responsibility after the choices have been made. Once again, my proposal can handle a coalition of agents, not just a single agent. Based on my notion of actual cause in situation calculus synchronous game structures, I define a new notion of causal responsibility, which, just like actual causation, is also an ex-post notion. I also propose a combined notion of causal and strategic responsibility. I show that causal, strategic, and combined notions of responsibility are extensionally distinct.

3. Properties of Actual Cause and Causal Responsibility:

   I prove a general property, showing that my definition of actual cause does not suffer from overdetermination, which happens when a subset of the identified causes would have been sufficient to bring about the effect. Finally, I also show that my causes do not suffer from the preemption problem, which happens when two competing events try to achieve the same effect, and the latter of these fails to do so, as the earlier one has already achieved the effect.

   On the responsibility side, I investigate the relationship between various notions of responsibility. I also study temporal consistency between ex-ante and ex-post notions of responsibility and list the conditions for the persistence of passive as well as combined responsibility.

4. Formal Examples:

   Aside from the novel theoretical framework, the thesis includes some formal examples based on the famous *bottle* example [4], which provide intuitive justification of my definition of actual cause and show that the account is indeed free of common problems such as preemption and over-determination.

   I also formalize an *attempted murder* scenario and illustrate multi-agent strategic responsibility in it. Furthermore, using this example, I show that strategic notions of responsibility cannot be used to capture all the subtleties of the domain, and causal and combined notions of responsibility indeed enrich the analysis.

1.4 Organization

The rest of this manuscript-style thesis is organized as follows.

- Chapter 2 – Literature Review.

  In this chapter, I present an overview of existing approaches to actual causation. The survey includes discussion on classical counterfactual methods [9, 2], structural-equation-based causal models [10, 3], and early action-theoretic frameworks [6, 11], with a specific focus on their limitations. In particular, I show that these formalizations have limited expressivity and cannot handle concurrent, multi-agent environments properly. Further, a survey of previous work on strategic responsibility and causal responsibility is also provided, outlining

the current landscape.

- Chapter 3 – Foundations.

  This chapter introduces the basic formal framework within which I develop the thesis. It details situation calculus, a second-order logic to reason about actions and dynamic domains, and discusses previous work on encoding a first-order variant of concurrent game structures within the situation calculus. It also details earlier attempts to define actual achievement causes in the situation calculus.

- Chapter 4 – On the Semantics of Actual Causality in situation calculus concurrent game structures.

  After defining the formal definitions, this chapter goes on to explain the definition and examination of refined causes and causal chains within synchronous game structures. Specific examples are provided to highlight how the approach can state all the different complete chains of causes and determine the minimal sets of moves that are necessary in each chain. Counterfactual analysis and being immune to over-determination and preemption are highlighted in order to illustrate the importance of the approach in multi-agent environments.

- Chapter 5 – Causal, Strategic, and Combined Responsibility Attribution in situation calculus concurrent game structures.

  In this chapter, I extend the framework developed in Chapter 4 to formalize

responsibility attribution in concurrent multi-agent settings. The discussion unifies causal and strategic perspectives on responsibility within the situation calculus synchronous game structures (SCSGS). I first define an account of actual causation in SCSGS that handles preemption and over-determination correctly, forming the causal basis for responsibility. Building on this, I introduce and compare distinct notions of responsibility—causal, strategic, and combined—showing that each captures a different mode of accountability in synchronous domains. The chapter also establishes key formal properties of these notions, including their persistence and mutual distinctness. Through formal examples, I demonstrate how causal reasoning complements strategic reasoning to yield a more complete understanding of multi-agent responsibility. Finally, I discuss the implications of this framework for explainability and accountability in autonomous systems and related domains.

- Discussion and Conclusion

  Finally, in this chapter, I summarize my contributions and point out the limitations of the proposal. I conclude the thesis by identifying potential future research directions.

Chapter 2

Literature Review

2.1 Introduction

This chapter surveys the main lines of work that are relevant to this thesis. I first review previous proposals on *actual causality*, beginning with Pearl's structural-equations model-based causal framework and its counterfactual foundations [10] , and then consider Halpern and Pearl's refinements that address preemption and over-determination [2]. These models provide the baseline for identifying minimal causes but remain limited in expressivity. I then discuss extensions that try to address some of the restrictions of causal models. After that, I briefly point out recent work on formalization of actual causality in action-theoretic domains.

Next, I turn to research on *responsibility*. Chockler and Halpern [12] propose a structural-model account of responsibility and blame based on degrees of causal contribution. Subsequent work studies responsibility attribution in Seeing to it that (STIT) logics [13, 14]. More recent work consider temporal logics and planning,

introducing ex-ante and ex post notions of responsibility anticipation and attribution.

Together, these reviews highlight both the strengths and the gaps of existing accounts of causality and responsibility. In the next chapter, I will give more details on some of these to establish the foundation upon which I develop my own framework of actual cause and responsibility in situation calculus concurrent game structures.

2.2 Causality

Pearl, the Structural Model Approach Pearl's **causal model** [10, 2] provides one of the most influential formalizations of causality. In this setting, the world is represented by a collection of structural equations. Variables are either exogenous, determined outside the model, or endogenous, whose values are specified within the model. Causal dependence is evaluated counterfactually: if changing X in the model would alter the value of Y, then X is considered a cause of Y. This captures the intuition of asking, "what would happen if X were different?"

David Hume's notion of *constant conjunction* holds that causation is nothing more than regularly repeated instance-pairs of a cause and effect—whenever the first occurs, the second follows. [15, 16] Pearl's counterfactual framework can be seen as a formal development of this idea: rather than simply observing an association, one asks whether φ would still hold were $\vec{X}$ forced to differ—thus restoring Hume's empirical insight into a more powerful structural-causal treatment.

Counterfactual reasoning remains central to contemporary accounts of causality.

Before introducing these, it is useful to recall how counterfactual causation is traditionally formalized in the structural-causal models of Halpern and Pearl [2, 3].

Causal Models A causal model is given by $M = \langle \mathcal{S}, \mathcal{F} \rangle$, where

- $\mathcal{S} = \langle U, V, R \rangle$ is a *signature*, consisting of a finite set U of exogenous variables, a finite set V of endogenous variables, and a mapping R assigning each variable $X \in U \cup V$ its range of values $R(X)$.
- $\mathcal{F}$ assigns a structural equation to each $X \in V$, written $X := f_X(\vec{U}, \vec{V})$, which determines the value of X.

The "But-For" Counterfactual Test. An early and widely used notion of causality is the *but-for test*, which Hume discussed [15], originating in philosophy. Intuitively, $\vec{X} = \vec{x}$ is a cause of φ if φ is true in the actual model, but would not hold had $\vec{X}$ been set differently. In the structural-equations account, a *causal model* $M = \langle \mathcal{S}, \mathcal{F} \rangle$ is evaluated at a *context* $\vec{u}$ (an assignment to the exogenous variables). A formula φ denotes an event about endogenous variables, typically a Boolean condition such as a fluent (or a conjunction of fluents) being true in the solution induced by $(M, \vec{u})$ [10, 2, 3]. An *intervention* on variables $\vec{X}$ to values $\vec{x}'$, written $\langle\langle \vec{X} \leftarrow \vec{x}' \rangle\rangle$, replaces the structural equations for $\vec{X}$ by constants $\vec{x}'$, yielding the intervened model in which all other equations are solved while holding $\vec{X} = \vec{x}'$ fixed [10, 2]. Thus, the

standard "but-for" counterfactual test is expressed as

$$(M, \vec{u}) \models \varphi \quad \text{and} \quad (M, \vec{u}) \models \langle\langle \vec{X} \leftarrow \vec{x'} \rangle\rangle \neg\varphi$$

However, for some alternative $\vec{x'} \neq \vec{x}$. While intuitive, this definition breaks down in cases of preemption and over-determination.

Preemption. Preemption arises when an action masks the causal role of another by occurring earlier. In the well-known bottle example [3, 17], both agents throw stones, but Suzy's reaches the bottle first. Under simple but-for analysis, neither throw qualifies as a cause, since removing one still leaves the other.

Over-Determination. Over-determination occurs when multiple independent actions are each sufficient to produce the effect. For example, in a voting scenario [3] where more than half of the votes are required to win the ballot, if eleven agents cast their votes and eight supported Suzy, a naive account might identify all eight votes as causes of Suzy winning the ballot. In fact, only six would have been sufficient. The Halpern–Pearl framework [2, 18, 3] handles this by selecting any minimal subset of six, while treating the rest as redundant under the contingency set $\vec{W}$, as discussed below.

Halpern and Pearl's Refinements Halpern and Pearl refined Pearl's definition to address such problematic cases [3]. Justified using a handful of examples [1], their frameworks capture actual causes even in examples involving preemption or over-determination. A classic case is the bottle example, where two agents, Suzy and Billy, throw stones, but only one shatters the bottle. Halpern and Pearl's definition correctly identifies Suzy's throw as the actual cause, while excluding Billy's preempted action [18, 3], although using flawed reasoning; see [20]. In the following, I will give their definition.

Actual Cause. Given a causal setting $(M, \vec{u})$, a candidate event $\vec{X} = \vec{x}$, and outcome φ, $\vec{X} = \vec{x}$ is an *actual cause* of φ if the following conditions hold based on Halpern and Pearl's framework [2, 3].

AC1 *(Actuality)* $(M, \vec{u}) \models (\vec{X} = \vec{x}) \wedge \varphi$, which states the candidate event $\vec{X} = \vec{x}$ and the outcome φ both hold in the actual causal setting $(M, \vec{u})$.

AC2 *(Counterfactual Dependence under Contingencies)* There exists a partition of the endogenous variables into $\vec{Z}$ and $\vec{W}$, with $\vec{X} \subseteq \vec{Z}$, and there exist alternative values $\vec{x}'$ and settings $\vec{w}$ such that when $\vec{X}$ is set to $\vec{x}'$ and $\vec{W}$ is set to $\vec{w}$, φ becomes false; moreover, when $\vec{X}$ is held at its actual value and any subset of $\vec{W}$ is set to $\vec{w}$, φ remains true.

AC3 *(Minimality)* The conjunction $\vec{X} = \vec{x}$ is minimal: no strict subset of $\vec{X}$ satisfies

[1] Such example-driven development has proven to be susceptible to counter-examples [19]. In fact, Halpern and Pearl proposed at least three definitions, each successive one to rule out newly identified counter-examples. Here, I will discuss the latest "modified" definition.

both AC1 and AC2.

Here, φ denotes the outcome of interest (typically a fluent or conjunction of fluents that holds in the actual scenario) and thus represents what one seeks to explain as resulting from the causal process.

In the bottle example, the above conditions **AC1**–**AC3** capture Suzy's throw as the actual cause by considering the contingency $\vec{W} = \{BT\}$ with $BT \leftarrow 0$, reflecting that Suzy's action preempted Billy's.

Despite this, the Halpern-Pearl proposal suffers from expressive limitations. For instance, one cannot distinguish the occurrence of an event from its effects. There is no notion of event ordering or concurrency; all events are simply assumed to have happened.[2] Again, all events are considered to be independent. Finally, as argued in [20], the formalization forces one to reason about impossible worlds (e.g., one where Billy threw but the stone did not hit the bottle, despite having the assumption of a perfect throw!).

To deal with these issues, others have recently expanded Halpern–Pearl's account with additional features, such as temporal aspect [21] continuous variables [22], and general notions of counterfactuals that are non-interventionist in nature [23].

Besides the structural-equations models-based accounts, there has also been work on actual causation in other frameworks, for example, in formal action-theories using

[2]Indeed, it is not clear how one can even model the fact that Suzy's throw was performed before Billy's.

the notions of counterfactual dependence and production [20, 24], in the situation calculus using first-principles reasoning [6]; in propositional non-monotonic logic frameworks [25, 26] and from non-counterfactual "regularity" perspectives, such as [27]. Of particular interest is the recent action-theoretic definition of actual cause proposed by Batusov, Soutchanski [6]. This account has been extended to address knowledge of causes and their dynamics [28], as well as the causes of intentions [29], which has proven useful in explaining agent behavior. This work has also been extended to deal with non-deterministic domains [30, 31]. Since I will use this theory as my base framework, I will discuss this in detail in the next chapter.

2.3 Causal Responsibility

Independently, researchers have also investigated how to attribute responsibility when an agent either brings about an effect or fails to prevent it. Chockler and Halpern introduced a causal-model account of **responsibility and blame** [12]. In their framework, the degree of responsibility depends on how many changes are required to make an agent's action necessary for the outcome. For example, if Suzy's throw alone shatters the bottle, her responsibility is maximal. If many agents throw stones, responsibility is distributed, since removing one action does not alter the outcome. Blame is then defined as responsibility combined with the agent's epistemic state, linking causal models with reasoning about knowledge.

Formally, given a causal model M and context $\vec{u}$, the *degree of responsibility* of a

setting $\vec{X} = \vec{x}$ for an outcome φ is defined as:

$$Resp_{(M,\vec{u})}(\vec{X} = \vec{x}, \varphi) = \frac{1}{k+1},$$

where k is the minimal number of additional variables that must be fixed so that changing $\vec{X}$ changes the truth of φ. If $\vec{X} = \vec{x}$ is not an actual cause, the degree of responsibility is 0. An *exclusive cause* arises when no further variables must be fixed, yielding responsibility of 1.

This account focuses primarily on distributing responsibility among causes by degree. In contrast, my thesis emphasizes the definition of responsibility itself rather than its degree, since causal contribution is not the only form of responsibility that matters in multi-agent systems.

2.4 Strategic Responsibility

An alternative line of work employs STIT logic (*Seeing To It That*) to analyse responsibility [32, 33, 34]. In this tradition, the primary focus is on what is often called *active responsibility*. Intuitively, an agent is actively responsible for an event when the agent has a choice among available actions and deliberately selects one that ensures the event takes place. The STIT logic lineage begins with the pioneering work of Belnap, Perloff, and Xu on "achievement" STIT, where agency is captured within branching-time models of choice [35]. In this framework, responsibility is expressed through statements of the form "the agent sees to it that φ," linking responsibility

directly to strategic control over outcomes.

The STIT framework provides a rich formal tool for connecting deliberation, choice, and counterfactual dependence, which means that if the agent had chosen differently, the outcome would have been different. Extensions of STIT incorporate notions of causal refinement [14, 36], highlighting that responsibility in multi-agent settings can involve considerations of fairness and collective accountability. In this sense, *active responsibility* is tied to the agent's ability to ensure φ through a suitable strategy.

By contrast, *passive responsibility* (also called *counterfactual responsibility*) concerns cases where an agent is responsible because they *failed to prevent* an outcome. This notion is rooted in Frankfurt's principle of alternative possibilities [37]: an agent is responsible for an outcome if the agent had the ability to avoid it but did not. Passive responsibility is therefore attributed not on the basis of producing the effect, but on the ability to avoid it in the relevant counterfactual sense.

Responsibility in Multi-Agent Strategic Settings. In concurrent multi-agent domains, responsibility must account for **strategies**. Yazdanpanah, Dastani, Alechina, and Logan [7] formalize group responsibility based on the strategic ability of coalitions to guarantee outcomes: a coalition C is responsible for φ if it possesses a strategy that ensures φ regardless of how the remaining agents behave. Building on this view, De Giacomo *et al.* [8] refine responsibility into *active*, *weak-passive*, and *strong-passive* variants; these distinctions will be used later in this thesis.

In situation calculus concurrent game structures (SCSGSs), responsibility is captured operationally through a coalition's ability to enforce or prevent outcomes across all possible continuations of play. Informally, a coalition C *can force* φ from situation s if every path consistent with its strategy makes φ true.

Forward (Anticipatory) vs. Backward (Attributive) Responsibility. Recent work further distinguishes between *responsibility anticipation* and *responsibility attribution.*

- **Anticipatory (Forward) Responsibility** evaluates responsibility *before* execution, at planning time: an agent is forward-responsible if, from the current situation, it is strategically committed to an outcome across all possible future continuations. Parker, Grandi, and Lorini formalize such anticipation in multi-agent planning and LTL_f settings [38, 8].

- **Attributive (Backward) Responsibility** evaluates responsibility *after* execution, given the actual course of events. Here, the question becomes whether the agent (or coalition) could have prevented the outcome by deviating at some earlier point [39, 7, 40].

Anticipation underpins attribution, meaning that if an agent knowingly proceeds into a state where it anticipates responsibility, then, after the outcome occurs, responsibility can be attributed to that agent [38, 8]. In chapter 5, I integrate these strategic, active, and passive responsibility models within situation calculus concurrent game

structures, and connect them to the causal framework developed in Chapter 3.4. This allows both *forward-looking* reasoning about who will be responsible and *backward-looking* reasoning about who is responsible in actual causal contribution.

2.5 Conclusion

In this chapter, I briefly reviewed the strands of work that are indirectly related to this thesis. On *actual causality*, I surveyed Pearl's causal model framework and its counterfactual basis, together with Halpern and Pearl's refinements that address preemption and over-determination [10, 2, 3]. These accounts provide principled criteria for identifying actual causes, but they are severely limited in expressivity. I also outlined other related work on the actual cause.

On *responsibility*, I examined causal definitions of responsibility and blame that quantify degrees of causal contribution [12], and strategic perspectives [32, 33, 34], that distinguish active and passive as well as ex-post attribution from ex-ante anticipation [8, 38]. Together, these frameworks clarify how accountability can be assigned, yet they do not fully capture responsibility in concurrent multi-agent settings or its interaction with actual causation. As mentioned in the next chapter, I give details of some of these accounts that form the foundations of my proposal.

Chapter 3

Foundations

3.1 Introduction

In this chapter, I present the foundations for the rest of my thesis. In particular, I begin by detailing the situation calculus, introduced by McCarthy and Hayes [41], a language for representing and reasoning about dynamic domains. I then introduce the situation calculus synchronous game structures (SCSGS) [42] in which, at each timestamp, multiple agents can make concurrent moves. Finally, I discuss previous work on **actual causality** within the situation calculus, outlining epistemic formula, defining **direct** and **indirect** causes, and addressing *preemption* [9].

3.2 Situation Calculus

I briefly recall core notions of the Situation Calculus, a sorted (mostly) first-order language for representing and reasoning about dynamic domains [43]. In this language, there are three sorts, situations, actions and a generic object sort. *Situations*

are first-order terms and denote histories of action. There is a distinguished initial situation S_0, representing the empty sequence of actions. Performing action a in situation s yields $do(a, s)$. Here, $do(a, s)$ is a special function capturing the successor situation to s after a has been performed. For example, executing a_1 then a_2 in S_0 yield the situation $do(a_2, do(a_1, S_0))$. *Actions* are also terms that evolve situations; For example, $drop(r, b)$ might be used to represent the robot r's action of dropping block b.*Fluents* describe situation-dependent properties and take a situation as their last argument. For instance, Broken(b, do(drop(r,b), S_0)) means that block b is broken after the robot r has dropped it in situation S_0. Fluents can be relational or functional. A reserved predicate $Poss(a, s)$ is used to represent that action a is physically executable in situation s. $s \sqsubset s'$ means that situation s' can be reached from situation s by performing one or more actions. $s \sqsubseteq s'$ is an abbreviation of $s \sqsubset s' \vee s = s'$. $s < s'$ means $s \sqsubset s' \wedge Executable(s')$, where $Executable(s')$ means all actions in the history of s' was possible when they were executed. $s \leq s'$ is defined as $s < s' \vee s = s'$.

In the situation calculus, the dynamics of the domain are specified using a basic action theory (BAT) [43], which has the following components.

1. Foundational Axioms: These uniquely characterize the structure of situations, the precedence operator $\sqsubset$, and specify that *do* is injective:

$$do(a_1, s_1) = do(a_2, s_2) \rightarrow a_1 = a_2 \wedge s_1 = s_2,$$

$$\forall P.P(S_0) \wedge \forall a, s.(P(s) \supset P(do(a, s))) \supset \forall s.P(s),$$

$$\neg s \sqsubset S_0,$$

$$s \sqsubset do(a, s') \equiv s \sqsubseteq s'.$$

2. Action Precondition Axioms: There is also an action-precondition axiom for each action to state when actions are executable:

$$Poss(a, s) \equiv \varphi_a(s).$$

Here φ_a(s) is a formula that is uniform in s, meaning that it has no occurrence of Poss, $\sqsubset$, or other situation terms besides s, and does not quantify over situations. For instance, Poss(throw(agt,x),s) $\equiv$ Holding(agt,x,s) says that the action $throw(agt, x)$, i.e., an agent *agt* throws an object x, is possible in situation s if and only if agt is already holding x in situation s.

3. Successor State Axioms (SSA): These are given one per fluent and specify exactly when the fluent changes value after an action has been performed:

$$F(\bar{x}, do(a, s)) \iff \gamma_F^+(\bar{x}, a, s) \vee \big(F(\bar{x}, s) \wedge \neg\gamma_F^-(\bar{x}, a, s)\big)$$

where γ_F^+ captures the conditions under which F becomes true and γ_F^- those under which it becomes false. Assume the fluent $Broken(x, s)$, illustrating that the object x is broken at situation s. Two actions affect this fluent: $drop(agt, x)$, which means the agent agt drops the object x, and $repair(agt, x)$, which stands for the action of repairing the object x by the agent agt. The successor-state axiom for this fluent is as follows:

$$Broken(x, do(a, s)) \equiv (a = drop(agt, x) \wedge Fragile(x, s)) \vee$$

$$(Broken(x, s) \wedge a \neq repair(agt, s))$$

The above axiom states that object x, is broken in the situation that results from performing action a in situation s, if and only if action a is agt dropping x and x is fragile, or the object was already broken in situation s and the agent did not perform the action of repairing it. Here, the fluent $Fragile(x, s)$ states that x is fragile in situation s.

4. Unique Names Axioms for Actions: These state that distinct action symbols denote distinct actions, ensuring unambiguous reference. These can be automatically generated. For instance, *pickup* and *drop* actions are two different actions as stated in the following:

$$pickup(agt, s) \neq drop(agt', s')$$

And two pickup actions are not the same if they do not involve the same agent or the same object:

$$pickup(agt, s) = pickup(agt', s') \supset (agt = agt' \wedge s = s')$$

5. Initial State Axioms: Assertions about fluents at S_0 encode the initial state of the domain. For example $\mathcal{D}_{S_0} \supset \neg Broken(x, S_0)$ means that object x is not broken in the initial situation. Note that the initial situation need not be completely specified, allowing us to capture incomplete knowledge.

Together, these axioms provide a complete description of how fluents evolve across situations, enabling us to determine whether a property holds after the execution of a sequence of actions. This is a central reasoning task in the situation known as the *projection problem*. Rather than unfolding situations into long action sequences, this reasoning is supported by *regression* [43]. Regression systematically rewrites a formula referring to a future situation $do(\alpha, s)$ back into an equivalent formula about the earlier situation s, using the successor state axioms to eliminate occurrences of fluents at later situations. In doing so, all reference to do is removed, yielding a formula that concerns only the initial situation S_0. Thus, proving whether a property holds after executing a complex action sequence reduces to checking a first-order entailment at S_0. This avoids the need for second-order frame axioms and provides a tractable, uniform method for reasoning about dynamic change.

3.3 Situation Calculus Concurrent Game Structures (SCSGS)

The standard situation calculus does not allow concurrent action occurrence. One can thus only model turn-taking multi-agent systems there in. Many domains, however, feature agents that act *synchronously*. I adopt situation calculus concurrent game structures (SCSGS) [42], which is a first-order extension of concurrent game structures used with logics such as alternating time temporal logic (ATL*). In addition, SCSGS uses an action theory to specify how the agent moves make the fluents change and address the frame problem. It extends the situation calculus so that, at each timestamp, each of the n fixed agents performs a move, which jointly forms a single action. In SCSGS, situations result from these *joint* moves. Fluents also depend on joint moves, and successor-state axioms are stated accordingly.

To support this, SCSGS introduces an explicit subsort of *agents* (a finite set $Ag_1, \ldots, Ag_n$). Each agent is denoted by a distinct constant, with unique-names axioms $ag_i \neq ag_j$ for $i \neq j$, and a domain-closure axiom stating that $agent(x)$ holds if and only if x is one of these constants, i.e., $agent(x) \equiv (x = ag_1 \vee \cdots \vee x = ag_n)$.

Another subsort of object called moves is used to denote agent moves. For this, I will use the function symbols $M_i(\vec{x})$ over objects $\vec{x}$:

$$Move(m) \equiv \bigvee_i \exists \vec{x}.\ m = M_i(\vec{x}).$$

Each agent may have many available moves in a given situation. Unique-names and domain-closure axioms ensure the distinctness of these moves.

$$M_i(\vec{x}) \neq M_j(\vec{y}) \quad \textit{for } i \neq j,$$

$$M_i(\vec{x}) = M_i(\vec{y}) \supset \vec{x} = \vec{y}.$$

At each timestamp there is exactly one *joint* action, the concurrent execution of one move per agent, written $tick(m_1, \ldots, m_n)$, where n is the fixed number of agents and each m_i is a *move* by agent i.

Legal moves. For each agent, availability of moves is specified using *LegalM*. For agent Ag_i and move M_i:

$$LegalM(Ag_i, M_i(\vec{x}), s) \stackrel{\text{def}}{=} \Phi_{Ag_i,M_i}(\vec{x}, s),$$

i.e., Ag_i is legally allowed to perform $M_i(\vec{x})$ in s iff the uniform formula $\Phi_{Ag_i,M_i}(\vec{x}, s)$ holds.

Besides legality of moves, SCSGS also defines preconditions of tick actions:

$$Poss(tick(m_1, \ldots, m_n), s) \equiv \wedge_{i=1,\ldots,n} LegalM(Ag_i, m_i, s).$$

Again, successor-state axioms are specified for each fluent. Finally, I also need to specify what is true initially in S_0.

In the next chapter, I develop a formalization of actual cause in SCSGS. But before that, let me briefly go over previous work on actual cause in the situation calculus.

3.4 Actual Causality in Situation Calculus

Actual Causality Batusov and Soutchanski [6] recently proposed a formalization of actual causality in the situation calculus. In their meta-theoretic formalization, they only considered sequences of actions as scenarios. They proposed a definition of achievement cause, where the effect was false initially, but became true afterwards, and one tries to figure out the actions that were responsible for achieving the effect; And a definition of maintenance cause, where the effect is considered to be true before and after the sequence of actions in the scenario occurred, and the goal is to determine what "maintained" it.

Khan and Lespérance [44] recently redirected Butusov and Soutchanski's proposal of achievement cause to embed it within the language of situation calculus and to study epistemics of causation and its dynamics. In the following, I describe the formalism, which I adopt in later chapters.

In this framework, causes are computed relative to a *causal setting*.

Definition 3.4.1 (Causal Setting)**.** *A causal setting $\mathcal{C}$ is a tuple $\langle \mathcal{D}, \sigma, \varphi \rangle$, where $\mathcal{D}$ is a basic action theory (BAT), σ is an executable situation, called the* scenario*, and φ is a situation-suppressed formula that is uniform in s, representing the effect that*

was observed after the actions in the scenario have been performed.

In the situation calculus, all effects are the result of named actions. Thus, causes of φ are actions from the scenario. To distinguish between multiple occurrences of the same action within a scenario, Khan and Lesperance used a notion of the timestamp of an action occurrence, which I also adopt

Definition 3.4.2 (Timestamp of a Situation)**.** *Given a situation σ, I define a function $timeStamp(\sigma)$ as:*

$$timeStamp(S_0) = 0,$$

$$\forall a, s. timeStamp(do(a, s)) = timeStamp(s) + 1.$$

This assigns a unique timestamp to each action occurrence within a scenario [3]. Note that one can also use the situation calculus where the action occurred for this purpose, but timestamps are more convenient in the context of causal knowledge and its possible worlds semantics, where these provide a rigid designator for all possible worlds.

In later developments, Khan and Lespérance extended this account to reason not only about what actually caused an effect, but also about an agent's knowledge of causal relations across possible worlds [44]. This is done using dynamic (situation-suppressed) formulae, which allow expressing how the truth of a proposition evolves along the scenario without explicitly indexing situations. These dynamic formulae

serve as the basis for evaluating causation within epistemic models, where different possible histories compatible with an agent's knowledge are considered. In this sense, the causal analysis becomes epistemic: it distinguishes what is an actual cause in the real scenario from what agents believe could have been a cause in alternative epistemically accessible situations. This dynamic formula machinery will be used later when I study how actual causality interacts with strategic responsibility in chapter 5.

I next give the definition of actual cause in the situation calculus, starting with direct/primary cause, which is the action that directly brings about the effect. For example, the shattering of a window might be directly caused by the throwing of a stone in some scenarios with that action.

Definition 3.4.3 (Primary Cause)**.**

$$\mathit{CausesDirectly}(a, ts, \varphi, s) \stackrel{\text{def}}{=} \exists s_a.\ \mathit{timeStamp}(s_a) = ts \wedge (S_0 < do(a, s_a) \leq s)$$

$$\wedge \neg\varphi[s_a] \wedge \forall s'.[do(a, s_a) \leq s' \leq s \supset \varphi[s']].$$

They also defined indirect causes, which do not produce φ themselves, but are necessary enablers for a direct cause—for instance, acquiring the stone needed to perform the throw might be an indirect cause of the window shattering. Formally:

An action a indirectly causes φ at timestamp ts if there is a timestamp ts' in future situations, and an action b that directly or indirectly causes φ, and either:

1. Action a makes performing action b possible or,

2. Without doing action a, φ would not hold after action b is executed.

Khan and Lesperance [28] also propose a definition for the actual cause that can compute both primary and indirect causes.

Definition 3.4.4 (Actual Cause)**.**

$$
\begin{aligned}
Causes(a, ts, \varphi, s) \stackrel{\text{def}}{=} \forall P.[&\forall a, ts, s, \varphi.(CausesDirectly(a, ts, \varphi, s) \supset P(a, ts, \varphi, s)) \\
&\wedge \forall a, ts, s, \varphi.(\exists a', ts', s'.(CausesDirectly(a', ts', \varphi, s) \\
&\quad \wedge timeStamp(s') = ts' \wedge s' < s \\
&\quad \wedge P(a, ts, [Poss(a') \wedge After(a', \varphi)], s') \\
&\quad \supset P(a, ts, \varphi, s)) \\
&] \supset P(a, ts, \varphi, s).
\end{aligned}
$$

This states that *Causes* is the least relation P closed under (i) inclusion of direct causes and (ii) propagation along enabling links: if a' at ts' is a primary cause of φ in s, s' precedes s with $timeStamp(s') = ts'$, and a at ts is a cause of $[Poss(a') \wedge After(a', \varphi)]$ in s' (ensuring a' is executable and yields φ), then $(a, ts, \varphi, s) \in P$. Here, $After(a', \varphi)$ means that after performing the action of a', φ becomes true.

Integrating these notions into the situation calculus yields a clear, compositional account of direct, indirect, and actual causation within executable scenarios, providing the formal basis used throughout this thesis.

3.5 Conclusion

This chapter established the formal groundwork for the thesis. I recalled the *situation calculus*, basic action theories in the situation calculus, and reasoning about projection using regression. I then introduced a first-order variant of concurrent game structures, called the *situation calculus concurrent game structures (SCSGS)*. I also introduced (epistemic) dynamic formula and presented the semantics of actual causality in the Situation Calculus. With these foundations in place, in the next chapter, I develop a theory of actual causation within SCSGS, followed by responsibility attribution that leverages this causal structure in synchronous multi-agent domains in chapter 5.

Chapter 4

On the Semantics of Actual Causality in Situation Calculus Concurrent Game Structures

Abstract

Key to the formalization of rationality is the study of actual causation. Halpern and Pearl's pioneering work on causal models is based on structural-equations models, which assumes an overly simplistic model of action and change. Although much recent work within action-theoretic frameworks has appeared to deal with this, all of these accounts share a common and strong limitation, that the scenario or history of actions in these are assumed to be linear sequences of actions or traces. To deal with this, in this paper I study causation in a synchronous game-theoretic logic framework that allows concurrent moves by multiple agents. my framework is based

on situation calculus concurrent game structures. I show that my formalization has some interesting properties and handles the issues associated with preemption and over-determination well.

4.1 Introduction

Actual causality, also known as token-level causality, is the problem of identifying the causes of an observed effect from a given history of events or actions (also called, the scenario) [1]. Based on David Hume's original proposal, this problem has been studied extensively both from counterfactual perspectives (e.g., [2, 3, 4, 5, 1, 6, 7, 8, 9, 10, 11, 12]) as well as from regularity approaches (e.g., [13, 14, 15]). The former involve studying causes by observing what would have happened had some of the events in the original scenario not occurred, while the latter accounts define causation from the observation that causes are regularly followed by their effects (one interpretation of this, among many, states that a cause is an insufficient but necessary part of a sufficient condition that is itself unnecessary for bringing about the observed effect [13]). Others have attempted to combine these two approaches [16, 17].

In recent years researchers have become increasingly interested in studying causation within more expressive action-theoretic frameworks, in particular in that of the situation calculus [18, 19, 16, 20, 21, 22]. In contrast to the popular structural equations models-based or SEM-based causal models [23], these are based on a formal theory of action, and thus incorporate important aspects such as action preconditions,

effects, and frame conditions, and temporal order of action occurrence into the model, allowing one to capture, e.g., non-persistent change supported by fluents and possible dependencies between events. Moreover, this allows one to formalize causation from the perspective of individual agents by defining a notion of epistemic causation [24] and by supporting causal reasoning about conative effects, which in turn has proven useful for explaining agent behaviour using causal analysis [25] and has the potential for defining important concepts such as responsibility and blame [26].

A major limitation of these proposals, however, is that they take the scenario to be a linear sequence of single-agent actions. In multi-agent settings, this means that they are restricted to turn-taking games. To overcome these limitations, in this paper, I consider causation in *multi-agent synchronous games*. My account is based on Situation Calculus Synchronous Game Structures (SCSGS) [27], where I have a single action *tick* whose effects depend on the combination of moves selected by the players. Each agent selects its move without knowing which move is selected by the other agents. As I will see, in domains with synchronous concurrency, besides the usual preemption problem,[1] I also face the problem of over-determination, as there may be more than one subset of the moves that are sufficient to cause the effect. In this paper, I extend previous accounts of actual causation in the situation calculus [19, 24] to identify minimal subsets of moves by some of the agents that are causes of the effect, i.e., sufficient to cause it. I also identify causal chains consisting of such

[1]Preemption happens when two competing events try to achieve the same effect, and the latter of these fails to do so, as the earlier one has already achieved the effect.

minimal sets of moves, and notice that there may be several of them in the scenario for a given effect. I show that my notion of causal chains handles the issues associated with preemption and over-determination well.

In the next section, I start by reviewing the situation calculus (SC) and SCSGS. I also present my running example. In Section 3, I show how previous work on actual causation in the SC can be modified to identify causal chains. In Section 4, I consider minimal sets of agent moves as causes and the associated causal chains. In Section 5, I present some properties of my formalization. I conclude with some discussion in Section 6.

4.2 Preliminaries

Situation Calculus (SC). The SC is a well-known second-order language for representing and reasoning about dynamic worlds [28, 29]. In the SC, all changes are due to named actions, which are terms in the language. Situations represent a possible world history resulting from performing some actions. The constant S_0 is used to denote the initial situation where no action has been performed yet. The distinguished binary function symbol $do(a, s)$ denotes the successor situation to s resulting from performing the action a. The expression $do([a_1, \cdots, a_n], s)$represents the situation resulting from executing actions $a_1, \cdots, a_n$, starting with situation s. As usual, a relational/functional fluent representing a property whose value may change from situation to situation takes a situation term as its last argument. There is a special

predicate $Poss(a, s)$ used to state that action a is executable in situation s. Also, the special binary predicate $s \sqsubset s'$ represents that s' can be reached from situation s by executing some sequence of actions. $s \sqsubseteq s'$ is an abbreviation of $s \sqsubset s' \vee s = s'$. $s < s'$ is an abbreviation of $s \sqsubset s' \wedge Executable(s')$, where $Executable(s)$ is defined as $\forall a', s'.\ do(a', s') \sqsubseteq s \supset Poss(a', s')$, i.e. every action performed in reaching situation s was possible in the situation in which it occurred. $s \leq s'$ is an abbreviation of $s < s' \vee s = s'$.

In the SC, a dynamic domain is specified using a basic action theory (BAT) $\mathcal{D}$ that includes the following sets of axioms: (i) (first-order or FO) initial state axioms $\mathcal{D}_{S_0}$, which indicate what was true initially; (ii) (FO) action precondition axioms $\mathcal{D}_{ap}$, characterizing $Poss(a, s)$; (iii) (FO) successor-state axioms $\mathcal{D}_{ss}$, indicating precisely when the fluents change; (iv) (FO) unique-names axioms $\mathcal{D}_{una}$ for actions, stating that different action terms represent distinct actions; and (v) (second-order or SO) domain-independent foundational axioms Σ, describing the structure of situations [30]. Although the SC is SO, Reiter [29] showed that for certain type of queries ϕ, $\mathcal{D} \models \phi$ iff $\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \models \mathcal{R}[\phi]$, where $\mathcal{R}$ is a syntactic transformation operator called *regression* and $\mathcal{R}[\phi]$ is a SC formula that compiles dynamic aspects of the theory $\mathcal{D}$ into the query ϕ. Thus reasoning in the SC for a large class of interesting queries can be restricted to entailment checking w.r.t a FO theory [29].

Synchronous Game Structures (SCSGS). Following [27], I focus on games where there are n players/agents each of whom chooses a move at every time step. All such

moves are executed *synchronously* and determine the next state of the game. At each time step, the state of the game is fully observable by all agents, as are all past moves of every agent. To represent such multi-player synchronous games, I use a special class of BATs, called *situation calculus synchronous game structures (SCSGSs)*, which are defined as follows.

Agents. A SCSGS D involves a finite set of n agents, and I use a subsort *agents* of *Objects* which includes these finitely many agents $Ag_1, \ldots, Ag_n$, each denoted by a constant, and for which unique names $Ag_i \neq Ag_j$ for $i \neq j$ and domain closure $agent(x) \equiv x = Ag_1 \vee \cdots \vee x = Ag_n$ hold.

Moves. I also use a second subsort *Moves* of *Objects*, representing the possible moves of the agents. These come in finitely many types, represented by function symbols $M_i(\vec{x})$, which are parameterized by objects $\vec{x}$, with $Move(m) \equiv \bigvee_i \exists \vec{x}.m = M_i(\vec{x})$. Given that the parameters range over *Objects*, each agent may have an infinite number of possible moves at each time step. I have unique name and domain closure axioms (parameterized by objects) for these functions $M_i(\vec{x}) \neq M_j(\vec{y})$ for $i \neq j$, and $M_i(\vec{x}) = M_i(\vec{y}) \supset \vec{x} = \vec{y}$.

Actions. In SCSGSs, there is only *one action type*, $tick(m_1, \ldots, m_n)$, which represents the execution of a joint move by all the agents at a given time step. The action *tick* has exactly n parameters, $m_1, \ldots, m_n$, one per agent, which are of sort *Moves* and corresponds to the simultaneous choice of the move to perform by the n different agents.

Legal moves. The *legal moves* available to each agent in a given situation are specified formally using a special predicate $LegalM$, which is defined by statements of the following form (one for each agent Ag_i and move type M_i): $LegalM(Ag_i, M_i(\vec{x}), s) \stackrel{\text{def}}{=} \Phi_{Ag_i,M_i}(\vec{x}, s)$, i.e., agent Ag_i can legally perform move $M_i(\vec{x})$ in situation s if and only if $\Phi_{Ag_i,M_i}(\vec{x}, s)$ holds. Technically $LegalM$ is an abbreviation for $\Phi_{Ag_i,M_i}(\vec{x}, s)$, which is a uniform formula (i.e., a formula that only refers to a single situation s).

Precondition axioms. The precondition axiom for the action *tick* is fixed and specified in terms of $LegalM$ as follows: $Poss(tick(m_1, \ldots, m_n), s) \equiv \wedge_{i=1,\ldots,n} LegalM(Ag_i, m_i, s)$. Thus the joint action by all agents $tick(m_1, \ldots, m_n)$ is executable if and only if each selected move m_i is a legal move for agent Ag_i in situation s. Since I only have one action type *tick*, this is the only precondition axiom in D_{poss}.

Successor state axioms. I have *successor state axioms* D_{ssa}, specifying the effects and frame conditions of the joint moves $tick(m_1, \ldots, m_n)$ on the fluents. Such axioms, as usual in basic action theories, are domain specific, and characterize the actual game under consideration. Within such axioms, the agent moves, which occur as parameters of *tick*, determine how fluents change as the result of joint moves.[2]

Initial situation description. Finally, the initial state of the game is axiomatized in the *initial situation description* D_{S_0} as usual, in a domain specific way.

Example. I use a variant of the well-known "bottle" example [3], where Suzy and

[2]In many cases, moves don't interfere with each other and the effects are just the union of those of each move. One can also exploit previous work on axiomatizing parallel actions to generate successor state axioms [29, 31].

Billy are throwing stones at a bottle. Suzy's stones are smaller and thus she requires two throws to break the bottle while Billy's stone is large and he needs just one throw to break it. The available moves of $ag \in \{Suzy, Billy\}$ can be one of $pick_{ag}$, representing the picking of one or more stones by agent ag (I assume that Suzy can pick both of her stones in one move), $throw_{ag}$, i.e. throwing of a stone by ag, and a catchall $other_{ag}$ move, denoting anything other than picking and throwing. The legality of these moves is specified below.

$$(a).\ LegalM(pick_{ag}, s) \stackrel{\text{def}}{=} \neg Holding(ag, s),$$

$$(b).\ LegalM(throw_{ag}, s) \stackrel{\text{def}}{=} Holding(ag, s), \qquad (c).\ LegalM(other_{ag}, s).$$

Thus, for example, throwing a stone is a legal move for agent ag in situation s if she is holding one or more stones in s. For simplicity, I assume that the $other_{ag}$ move is always possible.

There are three fluents in this domain, $Holding(ag, s)$, $Broken(s)$, and $SuzyThrown(s)$,

which means that the agent *ag* is holding their stones in situation s, the bottle is broken in s, and *Suzy* has already thrown once before in s, respectively. The successor-state axioms of these fluents are as follows.

$$(d).\ Holding(ag, do(a, s)) \equiv [ag = Suzy \wedge \exists m.\ a = tick(pick_{Suzy}, m)] \vee$$
$$[ag = Billy \wedge \exists m.\ a = tick(m, pick_{Billy})] \vee$$
$$[ag = Suzy \wedge Holding(ag, s) \wedge \neg(\exists m.\ a = tick(throw_{Suzy}, m) \wedge SuzyThrown(s))]$$
$$\vee[ag = Billy \wedge Holding(ag, s) \wedge \neg\exists m.\ a = tick(m, throw_{Billy})],$$
$$(e).\ Broken(do(a, s)) \equiv [Broken(s)] \vee [\exists m.\ a = tick(m, throw_{Billy})] \vee$$
$$[\exists m.\ a = tick(throw_{Suzy}, m) \wedge SuzyThrown(s)],$$
$$(f).\ SuzyThrown(do(a, s)) \equiv \exists m.\ a = tick(throw_{Suzy}, m) \vee SuzyThrown(s).$$

Thus, e.g., (d) states that an agent *ag* is holding stones after the action a is performed in situation s iff *ag* is Suzy and a is the *tick* action that involves her move of picking up stones; or if *ag* is Billy and a is the *tick* action that involves his move of picking up a stone; or *ag* is Suzy, who was already holding one or more stones in s, and a does not refer to a *tick* action that involves her move of throwing the last stone in hand; or *ag* is Billy, who already was holding a stone in s, and a is not a *tick* action involving his move of throwing a stone.

Finally, I assume that initially the agents are not holding any stones and the bottle

is not broken, as specified by the following initial state axioms:

$$(g).\ \forall ag.\ \neg Holding(ag, S_0), \qquad (h).\ \neg Broken(S_0).$$

I will use $\mathcal{D}_{bt}$ to refer to this axiomatization. □

4.3 Actual Causation in the SC

Based on Batusov and Soutchanski's original proposal [19], Khan and Lespérance (KL) recently defined achievement cause in the SC [24]. Both of these frameworks study achievement causation, i.e. assume that the effect is false initially and becomes true after the execution of the actions in the scenario. Also, both assume that the scenario is a linear sequence of actions, i.e. these do not allow concurrent actions.

To formalize reasoning about effects, KL [24] introduced the notion of *dynamic formulae*. An effect φ in their framework is thus a situation-suppressed dynamic formula.[3] Given an effect φ, the actual causes are defined relative to a scenario s. When s is ground, the tuple $\langle \varphi, s \rangle$ is often called a *causal setting* [19]. Also, it is assumed that s is executable, and φ was false before the execution of the actions in s, but became true afterwards, i.e. $\mathcal{D} \models Executable(s) \wedge \neg\varphi[S_0] \wedge \varphi[s]$. Here $\varphi[s]$ denotes the formula obtained from φ by restoring the appropriate situation argument into all fluents in φ (see Def. 4.3.2).

Note that since all changes in the SC result from actions, the potential causes of an

[3]While KL also study epistemic causation, I restrict my discussion to objective causality only.

effect φ are identified with a set of action terms occurring in s. However, since s might include multiple occurrences of the same action, I need a way to uniquely identify each action occurrence in the scenario s. To deal with this, KL required that each situation be associated with a time-stamp, which can then be used to uniquely identify an action occurrence. A time-stamp is an integer for their theory. KL assumed that the initial situation starts at time-stamp 0 and each action increments the time-stamp by one. Thus, their action theory includes the following axioms:

$$timeStamp(S_0) = 0, \qquad \forall a, s, ts.\ timeStamp(do(a, s)) = ts \equiv timeStamp(s) = ts - 1.$$

With this, causes in their framework is a non-empty set of action-time-stamp pairs.

The notion of *dynamic formulae* is defined as follows:

Definition 4.3.1. *Let $\vec{x}$, θ_a, and $\vec{y}$ respectively range over object terms, action terms, and object and action variables. The class of* dynamic formulae *φ is defined inductively using the following grammar:* $\varphi ::= P(\vec{x}) \mid Poss(\theta_a) \mid After(\theta_a, \varphi) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists\vec{y}.\ \varphi$.

That is, a dynamic formula (DF) can be a situation-suppressed fluent, a formula that says that some action θ_a is possible, a formula that some DF holds after some action has occurred, or a formula that can built from other DF using the usual connectives. Note that φ can have quantification over object and action variables, but must not include quantification over situations or ordering over situations (i.e. $\sqsubset$). I will use φ for DF.

$\varphi[\cdot]$ is defined as follows:

Definition 4.3.2.

$$\varphi[s] \stackrel{\text{def}}{=} \begin{cases} P(\vec{x}, s) & \text{if } \varphi \text{ is } P(\vec{x}) \\ Poss(\theta_a, s) & \text{if } \varphi \text{ is } Poss(\theta_a) \\ \varphi'[do(\theta_a, s)] & \text{if } \varphi \text{ is } After(\theta_a, \varphi') \\ \neg(\varphi'[s]) & \text{if } \varphi \text{ is } (\neg\varphi') \\ \varphi_1[s] \wedge \varphi_2[s] & \text{if } \varphi \text{ is } (\varphi_1 \wedge \varphi_2) \\ \exists\vec{y}.\ (\varphi'[s]) & \text{if } \varphi \text{ is } (\exists\vec{y}.\ \varphi') \end{cases}$$

I will now present a variant of KL's definition of causes in the SC. The idea behind how causes are computed is as follows. Given an effect φ and scenario s, if some action of the action sequence in s triggers the formula φ to change its truth value from false to true relative to $\mathcal{D}$, and if there are no actions in s after it that change the value of φ back to false, then this action is a *primary* or *direct* actual cause of achieving φ in s.

Definition 4.3.3 (Primary Cause [24])**.**

$$CausesDirectly(a, ts, \varphi, s) \stackrel{\text{def}}{=} \exists s_a.\ timeStamp(s_a) = ts \wedge (S_0 < do(a, s_a) \leq s) \\ \wedge \neg\varphi[s_a] \wedge \forall s'.(do(a, s_a) \leq s' \leq s \supset \varphi[s']).$$

That is, a executed at time-stamp ts is the *primary cause* of effect φ in situation s iff

a was executed in a situation with time-stamp ts in scenario s, a caused φ to change its truth value to true, and no subsequent actions on the way to s falsified φ.

Now, note that a (primary) cause a might have been non-executable initially. Also, a might have only brought about the effect conditionally and this context condition might have been false initially. Thus earlier actions in the trace that contributed to the preconditions and the context conditions of a cause must be considered as causes as well. The following definition captures this. It is as in [24], but here I extend it to specify the causal chain that links the cause to the effect. It captures both primary and indirect causes and specifies the causal chains. It defines $\mathit{CausesByChain}(a, ts, cc, \varphi, s)$, meaning that action a at timestamp ts is a cause of an effect φ in scenario s through causal chain cc: [4]

[4]In this, I need to quantify over situation-suppressed DF. Thus I must encode such formulae as terms and formalize their relationship to the associated SC formulae. This is tedious but can be done essentially along the lines of [32]. I assume that I have such an encoding and use formulae as terms directly.

Definition 4.3.4 (Actual Cause Through Causal Chain)**.**

$$
\begin{aligned}
&\mathit{CausesByChain}(a, ts, cc, \varphi, s) \stackrel{\text{def}}{=} \\
&\quad \forall P.[\forall a, ts, s, cc, \varphi.(\mathit{CausesDirectly}(a, ts, \varphi, s) \supset P(a, ts, ((a, ts)), \varphi, s)) \\
&\qquad \wedge \forall a, ts, cc', s, \varphi.(\exists a', ts', s'.(\mathit{CausesDirectly}(a', ts', \varphi, s) \\
&\qquad\quad \wedge \mathit{timeStamp}(s') = ts' \wedge s' < s \\
&\qquad\quad \wedge P(a, ts, cc', [\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)], s') \\
&\qquad\quad \wedge cc = \mathit{Append}(cc', (a', ts')) \\
&\qquad \supset P(a, ts, cc, \varphi, s)) \\
&\quad] \supset P(a, ts, cc, \varphi, s).
\end{aligned}
$$

Thus, *CausesByChain* is defined to be the least relation P such that if a executed at time-stamp ts directly causes φ in scenario s then (a, ts, cc, φ, s) is in P, where $cc = ((a, ts))$; and if a' executed at ts' is a direct cause of φ in s, the time-stamp of s' is ts', $s' < s$, and $(a, ts, cc', [\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)], s')$ is in P (i.e. a executed at ts is a direct or indirect cause of $[\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)]$ in s' through causal chain cc'), then (a, ts, cc, φ, s) is in P, where $cc = \mathit{Append}(cc', (a', ts'))$. Here the effect $[\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)]$ requires a' to be executable and φ to hold after a'. Also, *Append* is defined as follows.

Definition 4.3.5 (Append)**.**

$$
\mathit{Append}(((a_1, ts_1), \ldots, (a_n, ts_n)), (a, ts)) \stackrel{\text{def}}{=} ((a_1, ts_1), \ldots, (a_n, ts_n), (a, ts)).
$$

Tick Actions as Causes in the SCSGS. The above formalization of actual causation was formulated for domains specified by BATs in the situation calculus. However, it can be used directly for SCSGS domains, as long as one focuses on identifying the *tick* actions in the scenario that caused the effect, and causal chains consisting of *tick* actions. This is not surprising as SCSGS are special kinds of BATs. I illustrate this in the example below.

Example (cont'd). Consider the scenario σ_1, where: $\sigma_1 = do([tick(pick_{Suzy}, other_{Billy}), tick(throw_{Suzy}, pick_{Billy}), \quad tick(other_{Suzy}, other_{Billy}), \quad tick(throw_{Suzy}, throw_{Billy})], S_0)$. I want to find the actual causes of the effect $\varphi_1 = Broken(s)$. Figure 4.1 shows each action at each time-stamp of this scenario.

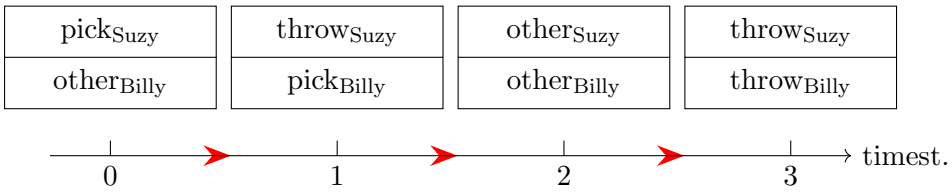

Figure 4.1: Concurrent stone throwing of Suzy and Billy.

I can show that:[5]

[5]Note that since Definition 4.3.4 inductively constructs the causal chain, considering some of the causes, e.g., the primary cause, will only give us a suffix of the complete causal chain *cc*; for simplicity, I thus only show the most indirect cause below, which captures the complete chain *cc*.

Proposition 1 (Complete Causal Chain in σ_1)**.**

$$\mathcal{D}_{bt} \models \mathit{CausesByChain}(\mathit{tick}(\mathit{pick}_{\mathit{Suzy}}, \mathit{other}_{\mathit{Billy}}), 0, cc, \varphi_1, \sigma_1),$$

$$\mathit{where}\ cc = ((\mathit{tick}(\mathit{pick}_{\mathit{Suzy}}, \mathit{other}_{\mathit{Billy}}), 0), (\mathit{tick}(\mathit{throw}_{\mathit{Suzy}}, \mathit{pick}_{\mathit{Billy}}), 1),$$

$$(\mathit{tick}(\mathit{throw}_{\mathit{Suzy}}, \mathit{throw}_{\mathit{Billy}}), 3)).$$

Explaining backward in *cc*, the last *tick* action executed at time-stamp 3 is included in the causal chain as (either of the moves in) it directly caused the breaking of the bottle. The second *tick* action executed at 1 is also included because it is a (secondary/indirect) cause as it brought about the preconditions of the last *tick* action (by making Billy's throw legal), besides bringing about the context condition (that *SuzyThrown*) under which Suzy's second throw can brake the bottle. Finally, the first *tick* action is also a cause as it made the second *tick* action executable. □

While the above formalization provides some insight on what *tick* actions are causes and can be used to identify the completely irrelevant *tick* actions, e.g. the one at time-stamp 2, observe that some irrelevant moves might still be included in the discovered causes, such as $\mathit{other}_{\mathit{Billy}}$ at time-stamp 0 in my example. In other words, my formalization of this does not specify what moves within the identified *tick* actions are contributing to the effect. To deal with this, I next propose a formalization of agent moves as causes.

4.4 Agent Moves as Causes in the SCSGS

I now go a step further by pinpointing the moves that actually contributed to the effect within the *tick* actions that are identified as causes. Note that, since unlike actions, agent moves within each *tick* action are concurrently performed, it is possible that more than one alternative chain of subsets of moves in the scenario are each by itself sufficient to bring about the effect. For instance, in my example, either $((pick_{Suzy}, 0), (throw_{Suzy}, 1), (throw_{Suzy}, 3))$ or $((pick_{Billy}, 1), (throw_{Billy}, 3))$ would have been sufficient to break the bottle. Just as with causal chains in the SC, I will identify these refined causal chains in two steps. In the first step, I identify the minimal set of moves in each action that is a direct cause of the effect in some refined chain (e.g., $throw_{Billy}$ in the last *tick* of the second refined causal chain above). I call these sets of direct causes *minimal moves primary causes* since they are minimal sets of moves that are causes. However, I must consider that there might be more than one minimal moves primary cause in one single action. For example in the last tick of my example, I can consider either only Suzy's throwing or Billy's throwing to be a minimal moves primary cause. In the second step, using refined causes, I define the refined chains mentioned above, which I call *minimal moves causal chains*.

In keeping with the formalization of dynamic domains in the SC, I will consider actions (but not moves) as (refined) causes.[6] Thus my formalization of this

[6]Note that the action theory specifies how the situation changes when actions, i.e., joint moves, are performed. This allows interfering or synergic effects to be specified.

does not omit the irrelevant moves in each time-stamp altogether, but rather replaces them with the special move *wait* within the *tick* action to remove their effects. *wait* has no effects (the domain modeler must ensure this) and is always legal: $\forall ag, s.\ LegalM(ag, wait, s)$. Thus, for instance, in my example, one such refined action that is a cause is $tick(pick_{Suzy}, wait)$. I collect all of these for all causal chains in my new definition of causes.

I now define minimal moves primary causes:

Definition 4.4.1 (Minimal Moves Primary Cause)**.**

$$MinMovCausesDir(a, ts, (a', ts), \varphi, s) \stackrel{\text{def}}{=} \exists s_a.\ timeStamp(s_a) = ts \wedge (S_0 < do(a, s_a) \leq s) \wedge$$
$$\neg\varphi[s_a] \wedge \forall s'.(do(a, s_a) \leq s' \leq s \supset \varphi[s']) \wedge MinSuffSubset(a', a, s_a, \varphi, s).$$

The above definition is exactly as the definition of primary causes (Def. 4.3.4), but has an additional parameter (a', ts) that returns a tuple consisting of a minimal subset of moves a' of the primary cause a that, when executed in s_a (i.e., the situation where the primary cause was executed in the original scenario), is sufficient to cause the effect φ in s (formalized using *MinSuffSubset* below), and the timestamp ts of a.

$MinSuffSubset(a', a, s', \varphi, s)$ means that the *tick* action a' consists of a sufficient subset of moves of a in s' to achieve φ up to s, and it is minimal.

Definition 4.4.2.

$$MinSuffSubset(a', a, s', \varphi, s) \stackrel{\text{def}}{=}$$

$$SuffSubset(a', a, s', \varphi, s) \wedge \neg\exists a^*.a^* \neq a' \wedge SuffSubset(a^*, a', s', \varphi, s).$$

a' is a sufficient subset of moves of a in s' to achieve φ up to s, i.e. $SuffSubset(a', a, s', \varphi, s)$, iff a' is a subset of moves of a, and the execution of this subset a' in s' is sufficient to cause φ in s, i.e. φ becomes true after a' is executed in s' and it remains true after the subsequent actions between $do(a, s')$ and s are executed starting in $do(a', s')$.

Definition 4.4.3.

$$\begin{aligned} SuffSubset(a', a, s', \varphi, s) \stackrel{\text{def}}{=} & \; SubsetMovs(a', a) \wedge \exists s''.\ timeStamp(s'') = timeStamp(s) \wedge s' < s'' \\ & \wedge \forall a_1, s_1, a_1', s_1'.[(do(a, s') < do(a_1, s_1) \leq s \wedge do(a', s') < do(a_1', s_1') \leq s'' \wedge \\ & \qquad timeStamp(s_1') = timeStamp(s_1)) \supset (a_1 = a_1')] \\ & \wedge (\forall s_1'.(s' < s_1' \leq s'') \supset \varphi[s_1']). \end{aligned}$$

Here $SubsetMovs(a', a)$, meaning that the *tick* action a' is exactly as a, but with some of the moves possibly replaced with the *wait* move, is defined as follows:

Definition 4.4.4 (The Moves of a' Consists of a Subset of the Moves of a)**.**

$$\begin{aligned} SubsetMovs(a', a) \stackrel{\text{def}}{=} & \; \exists m_1', \cdots, m_n', m_1, \cdots, m_n.\ a' = tick(m_1', \cdots, m_n') \\ & \wedge a = tick(m_1, \cdots, m_n) \wedge (\forall j.\ 1 \leq j \leq n \supset (m_j' = m_j \vee m_j' = wait)). \end{aligned}$$

Using minimal moves primary causes, I next formalize minimal moves causal

chains. I do this by defining a variant of *CausesByChain* that inductively constructs the minimal moves chain instead.

Definition 4.4.5 (Minimal Moves Cause Through Causal Chain)**.**

$$
\begin{aligned}
&\mathit{MinMovesCausesByChain}(a, ts, cc, \varphi, s) \stackrel{\text{def}}{=} \\
&\quad \forall P.[\forall a, ts, cc, a', s, \varphi.(\mathit{MinMovCausesDir}(a, ts, (a', ts), \varphi, s) \supset P(a, ts, ((a', ts)), \varphi, s)) \\
&\qquad \wedge \forall a, ts, cc', s, \varphi.(\exists a'', a', ts', s'.(\mathit{MinMovCausesDir}(a', ts', (a'', ts'), \varphi, s) \\
&\qquad\quad \wedge \mathit{timeStamp}(s') = ts' \wedge s' < s \\
&\qquad\quad \wedge P(a, ts, cc', [\mathit{Poss}(a'') \wedge \mathit{After}(a'', \varphi)], s') \\
&\qquad\quad \wedge cc = \mathit{Append}(cc', (a'', ts')) \\
&\qquad \supset P(a, ts, cc, \varphi, s)) \\
&\quad] \supset P(a, ts, cc, \varphi, s).
\end{aligned}
$$

Thus *MinMovesCausesByChain* is the smallest set such that if a *tick* action a executed at time-stamp ts directly caused the effect φ in scenario s with the minimal moves primary cause (a', ts), then (a, ts, cc, φ, s) is in that set, where $cc = ((a', ts))$; and if a' executed at ts' is a direct cause of φ in s with minimal moves primary cause (a'', ts'), the time-stamp of s' is ts', $s' < s$, and $(a, ts, cc', [\mathit{Poss}(a'') \wedge \mathit{After}(a'', \varphi)], s')$ is in P (i.e. a executed at ts is a direct or indirect cause of $[\mathit{Poss}(a'') \wedge \mathit{After}(a'', \varphi)]$ in s' through minimal moves causal chain cc'), then (a, ts, cc, φ, s) is in P, where $cc = \mathit{Append}(cc', (a'', ts'))$. This thus incrementally constructs the refined causal chains using *MinMovCausesDir*.

Example (cont'd). I can show the following result about refined causal chains.[7]

Proposition 2 (Complete Refined Causal Chains in σ_1)**.**

$$\mathcal{D}_{bt} \models \mathit{MinMovesCausesByChain}(\mathit{tick}(\mathit{pick}_{\mathit{Suzy}}, \mathit{other}_{\mathit{Billy}}), 0, cc_1, \varphi_1, \sigma_1)$$

$$\wedge\, \mathit{MinMovesCausesByChain}(\mathit{tick}(\mathit{throw}_{\mathit{Suzy}}, \mathit{pick}_{\mathit{Billy}}), 1, cc_2, \varphi_1, \sigma_1), \;\; \textit{where}$$

$$cc_1 = ((\mathit{tick}(\mathit{pick}_{\mathit{Suzy}}, \mathit{wait}), 0), (\mathit{tick}(\mathit{throw}_{\mathit{Suzy}}, \mathit{wait}), 1), (\mathit{tick}(\mathit{throw}_{\mathit{Suzy}}, \mathit{wait}), 3)),$$

$$\textit{and } cc_2 = ((\mathit{tick}(\mathit{wait}, \mathit{pick}_{\mathit{Billy}}), 1), (\mathit{tick}(\mathit{wait}, \mathit{throw}_{\mathit{Billy}}), 3)).$$

Thus, in my example, I have two distinct causal chains, one that stems from the refined primary cause involving Suzy's second throw move and Billy's *wait* move at time-stamp 3, and another originating from Billy's throw and Suzy's *wait* move, again at time-stamp 3. Importantly, as I will show in the next section, there is no causal chain that involves both Suzy and Billy's throw moves at time-stamp 3, avoiding the problem of over-determination (as each of these moves would have been sufficient for breaking the bottle).

Note that in the above, all the minimal move causes involve a single (non-*wait*) move. But this is not always the case. For example, if I replace the effect φ_1 by $\varphi^* = \mathit{Broken}(s) \wedge \mathit{SuzyThrown}(s)$, then a minimal move causal chain is:

$((\mathit{tick}(\mathit{pick}_{\mathit{Suzy}}, \mathit{wait}), 0), (\mathit{tick}(\mathit{throw}_{\mathit{Suzy}}, \mathit{pick}_{\mathit{Billy}}), 1), (\mathit{tick}(\mathit{wait}, \mathit{throw}_{\mathit{Billy}}), 3))$.

[7] Again, as in Proposition 1, I choose the most indirect causes to show the complete refined causal chains.

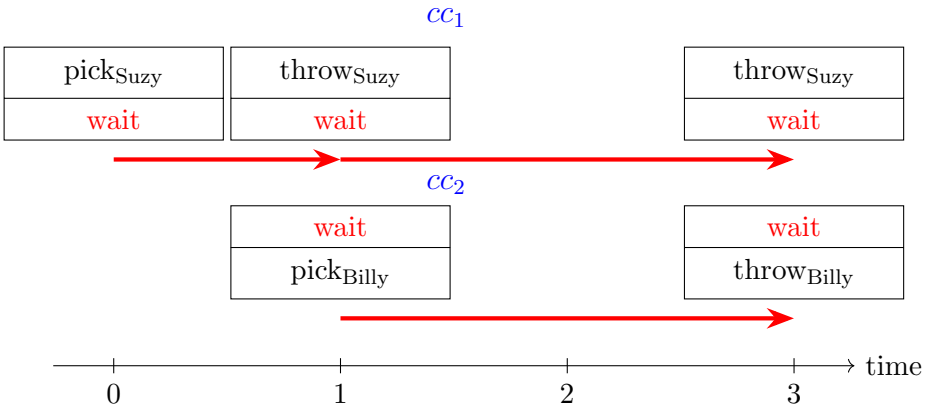

Figure 4.2: Refined causal chains cc_1 (Suzy's moves) and cc_2 (Billy's moves).

4.5 Properties

I now show that my formalization of refined causes and refined causal chains have some interesting properties. I start with the problem of preemption.

Preemption. Preemption occurs when there is more than one competing contributor (actions) to an effect, but they happen one after another/consecutively. In such cases, only the first of these should be identified as the actual cause. The (effects of the) latter actions are said to be preempted by the actual cause. My definition of refined causes and causal chains above is based on Khan and Lespérance's formalization of causes [24], which filter out the preempted contributors, and thus my formalization also handles the preemption problem correctly. I illustrate this using the following example.

Example 2. Consider the new scenario σ_2, where

$$\sigma_2 = do([tick(pick_{Suzy}, pick_{Billy}), tick(throw_{Suzy}, other_{Billy}), tick(throw_{Suzy}, other_{Billy}),$$
$$tick(other_{Suzy}, throw_{Billy})], S_0).$$

In this, I can show the following result.

Proposition 3.

$$\mathcal{D}_{bt} \models \neg\exists cc.\ CausesByChain(tick(other_{Suzy}, throw_{Billy}), 3, cc, \varphi_1, \sigma_2)$$
$$\wedge \neg\exists cc, m.\ MinMovesCausesByChain(tick(m, throw_{Billy}), 3, cc, \varphi_1, \sigma_2).$$

Thus as expected, Billy's throw is not considered as part of any (refined) causal chain.

Over-determination. Over-determination happens when the effect is contributed by some events, but a smaller subset of these would have been sufficient for the effect to hold. For example, in a voting scenario, where 6 out of 10 votes are required for a candidate to win, if 7 voters voted for candidate A, saying that all of these 7 votes are the cause of candidate A's winning the ballot would be over-determination as any 6 of these would suffice. An acceptable solution to this is to only identify any 6-vote subsets to be an actual cause [23] (or refined causal chain, in my formalization).

Example (cont'd). Using my first bottle example where the scenario is σ_1, I now argue that my notion of refined causal chains avoids over-determining causes. Note that

in this example, my *MinMovesCausesByChain* construct avoids over-determination by inductively specifying all possible sets of refined causal chains that are sufficient to break the bottle. As discussed above, there are only two refined causal chains; the first one only consists of Billy's moves, and the second only consists of Suzy's. The definition *MinMovesCausesByChain* ensures that only these two chains exist, since if I were to consider the chain that includes the time-stamp 3 throw moves of both Suzy and Billy, it will not be a minimal one as either throw would have been sufficient to break the bottle. Hence, I cannot have the moves of both agents involved in the same refined causal chain. Formally, I can show that:

Proposition 4.

$$\mathcal{D}_{bt} \models \neg\exists a, ts, cc.\ \mathit{MinMovesCausesByChain}(a, ts, \\ \mathit{Append}(cc, (\mathit{tick}(\mathit{throw}_{\mathit{Suzy}}, \mathit{throw}_{\mathit{Billy}}), 3)), \varphi_1, \sigma_1).$$

Thus, $\mathit{tick}(\mathit{throw}_{\mathit{Suzy}}, \mathit{throw}_{\mathit{Billy}})$ at time-stamp 3 cannot be a part of any causal chain for any action and timestamp.

In general, since refined causal chains are constructed to be minimal, if I have two refined chains in the same timestamp, it cannot be the case that one of them is a refined version of the other (i.e., has a subset of moves of the other).

Theorem 4.5.1 (No Over-Determination)**.**

$$\mathcal{D} \models \forall a, a_1, a_2, ts, cc, \varphi, s.(\mathit{MinMovesCausesByChain}(a, ts, \mathit{Cons}((a_1, ts), cc), \varphi, s) \wedge$$
$$\mathit{MinMovesCausesByChain}(a, ts, \mathit{Cons}((a_2, ts), cc), \varphi, s)$$
$$\supset (a_1 = a_2 \vee (\neg \mathit{SubsetMovs}(a_1, a_2) \wedge \neg \mathit{SubsetMovs}(a_2, a_1))),$$

where $\mathit{Cons}((a, ts), cc)$ *denotes the sequence where* (a, ts) *is added to the front of* cc*.*

Proof Sketch: By induction on the length of the minimal moves causal chain using properties of minimal sufficient subsets of joint moves. □

Persistence. Finally, I study the conditions under which (refined) causal chains persist when the scenario changes.

Theorem 4.5.2 (Persistence of the Causal Chain)**.**

$$\mathcal{D} \models \forall s, s', cc, ts, a, \varphi.\ \mathit{CausesByChain}(a, ts, cc, \varphi, s) \wedge s \leq s'$$
$$\wedge \forall s^*, a^*.(s \leq do(a^*, s^*) \leq s' \supset \varphi[s^*])$$
$$\supset \mathit{CausesByChain}(a, ts, cc, \varphi, s').$$

Proof Sketch: By induction on the length of the causal chain. □

That is, if a *tick* action a executed in ts is the cause of an effect φ in scenario s through causal chain cc, then a in ts remains the cause of φ in all subsequent situations/scenarios s' if φ does not change after it was achieved in s. This is because since the situation where φ was achieved does not change in the extended scenario,

neither does the causal chain.

A similar result can be shown for refined causal chains.

Theorem 4.5.3 (Persistence of Refined Causal Chains)**.**

$$\mathcal{D} \models \forall s, s', cc, ts, a, \varphi.\ MinMovesCausesByChain(a, ts, cc, \varphi, s) \wedge s \leq s'$$
$$\wedge \forall s^*, a^*.(s \leq do(a^*, s^*) \leq s' \supset \varphi[s^*])$$
$$\supset MinMovesCausesByChain(a, ts, cc, \varphi, s').$$

Proof Sketch: By induction on the length of the minimal moves causal chain using properties of minimal sufficient subsets of joint moves. □

4.6 Discussion and Conclusion

To support casual reasoning in multiagent domains, in this paper I proposed a formalization of actual causation in SCSGSs. I showed that one can adopt the definition of achievement causes in the SC to identify the relevant *tick* actions as causes and the associated causal chain in SCSGSs. However, since the chain can now include multiple sets of causes, each of which are independently sufficient to bring about the effect, to avoid over-determination I also formalized refinements of the identified causal chains. As shown, my account properly handles the problems with preemption and over-determination.

Note that, while the popular SEM-based frameworks of actual causation and their

derivatives seem to allow actions by multiple agents, these have other major limitations. For instance, often these consider the occurrence of a set of events without specifying their order of execution or whether they occur concurrently (all events are simply assumed to have happened). Events are also considered to be independent, which is another strong assumption. One cannot distinguish between the occurrence of an event and its effect holding. While recent action-theoretic formalizations of causality are meant to deal with these limitations, as mentioned earlier, these only handle single-agent or turn-taking multi-agent scenarios. Indeed to the best of my knowledge, my proposal is the first to accommodate synchronous concurrent moves by multiple agents while studying actual causes in formal action-theoretic frameworks. I am currently investigating how various notions of responsibility can be formalized based on this framework.

Acknowledgements

This work is partially supported by the National Science and Engineering Research Council of Canada, by the University of Regina, and by York University.

Chapter 4 References

[1] Joseph Y. Halpern. "Axiomatizing Causal Reasoning". In: *Journal of Artificial Intelligence Research* 12 (2000), pp. 317–337.

[2] David Lewis. "Causation". In: *Journal of Philosophy* 70.17 (1973), pp. 556–567.

[3] Ned Hall. "Two Concepts of Causation". In: *Causation and Counterfactuals*. Ed. by John Collins, Ned Hall, and L. A. Paul. MIT Press, 2004, pp. 225–276.

[4] Judea Pearl. *On the Definition of Actual Cause*. Tech. rep. R-259. University of California Los Angeles, 1998.

[5] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[6] Christopher Hitchcock. "The Intransitivity of Causation Revealed in Equations and Graphs". In: *The Journal of Philosophy* 98.6 (2001), pp. 273–299.

[7] Thomas Eiter and Thomas Lukasiewicz. "Complexity Results for Structure-based Causality". In: *Artificial Intelligence* 142.1 (2002), pp. 53–89.

[8] Alexander Bochman. "A Logic For Causal Reasoning". In: *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence,*

Acapulco, Mexico, August 9-15, 2003. Ed. by Georg Gottlob and Toby Walsh. Morgan Kaufmann, 2003, pp. 141–146.

[9] Mark Hopkins. "The Actual Cause: From Intuition to Automation". PhD thesis. University of California Los Angeles, 2005.

[10] Mark Hopkins and Judea Pearl. "Causality and Counterfactuals in the Situation Calculus". In: *Journal of Logic and Computation* 17.5 (2007), pp. 939–953.

[11] Joseph Y. Halpern. "A Modification of the Halpern-Pearl Definition of Causality". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. Ed. by Qiang Yang and Michael J. Wooldridge. AAAI Press, 2015, pp. 3022–3033.

[12] Maksim Gladyshev, Natasha Alechina, Mehdi Dastani, Dragan Doder, and Brian Logan. "Dynamic Causality". In: *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*. Ed. by Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu. Vol. 372. Frontiers in Artificial Intelligence and Applications. IOS Press, 2023, pp. 867–874.

[13] John Leslie Mackie. "Causes and Conditions". In: *American Philosophical Quarterly* 2.4 (1965), pp. 245–264.

[14] Richard W. Wright. “Causation in tort law”. In: *California Law Review* 73.6 (1985), pp. 1735–1828.

[15] Alexander Bochman. “Actual Causality in a Logical Setting”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Ed. by Jérôme Lang. ijcai.org, 2018, pp. 1730–1736.

[16] Shakil M. Khan and Mikhail Soutchanski. “Necessary and Sufficient Conditions for Actual Root Causes”. In: *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*. Ed. by Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang. Vol. 325. Frontiers in Artificial Intelligence and Applications. IOS Press, 2020, pp. 800–808.

[17] Sander Beckers and Joost Vennekens. “A Principled Approach to Defining Actual Causation”. In: *Synthese* 195.2 (2018), pp. 835–862.

[18] Vitaliy Batusov and Mikhail Soutchanski. “Situation Calculus Semantics for Actual Causality”. In: *Proceedings of the Thirteenth International Symposium on Commonsense Reasoning, COMMONSENSE 2017, London, UK, November 6-8, 2017*. Ed. by Andrew S. Gordon, Rob Miller, and György Turán. Vol. 2052. CEUR Workshop Proceedings. CEUR-WS.org, 2017.

[19] Vitaliy Batusov and Mikhail Soutchanski. "Situation Calculus Semantics for Actual Causality". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018, pp. 1744–1752.

[20] Asim Mehmood and Shakil M. Khan. "Towards a Definition of Primary Cause in Hybrid Dynamic Domains". In: *Proceedings of the 37th Canadian Conference on Artificial Intelligence (Canadian AI-24), Guelph, Ontario, Canada*. 2024.

[21] Maryam Rostamigiv, Shakil M. Khan, Yves Lespérance, and Mriana Yadkoo. "A Logic of Actual Cause for Non-Deterministic Dynamic Domains". In: *Proceedings of the 21st European Conference on Multi-Agent Systems (EUMAS-24), August 26-28, Dublin, Ireland*. Springer, 2024.

[22] Shakil M. Khan, Yves Lespérance, and Maryam Rostamigiv. "Reasoning about Actual Causes in Nondeterministic Domains". In: *Proceedings of the 39th Annual AAAI Conference on Artificial Intelligence (AAAI-25), February 25 - March 4, 2025, Philadelphia, Pennsylvania, USA*. AAAI Press, 2025.

[23] Joseph Y. Halpern. *Actual Causality*. MIT Press, 2016. ISBN: 978-0-262-03502-6.

[24] Shakil M. Khan and Yves Lespérance. "Knowing Why - On the Dynamics of Knowledge about Actual Causes in the Situation Calculus". In: *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems,*

Virtual Event, United Kingdom, May 3-7, 2021. Ed. by Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé. ACM, 2021, pp. 701–709.

[25] Shakil M. Khan and Maryam Rostamigiv. "On Explaining Agent Behaviour via Root Cause Analysis: A Formal Account Grounded in Theory of Mind". In: *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland*. Ed. by Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu. Vol. 372. Frontiers in Artificial Intelligence and Applications. IOS Press, 2023, pp. 1239–1247.

[26] Vahid Yazdanpanah, Enrico H. Gerding, Sebastian Stein, Mehdi Dastani, Catholijn M. Jonker, Timothy J. Norman, and Sarvapali D. Ramchurn. "Reasoning about responsibility in autonomous systems: challenges and opportunities". In: *AI Soc.* 38.4 (2023), pp. 1453–1464.

[27] Giuseppe De Giacomo, Yves Lespérance, and Adrian R. Pearce. "Situation Calculus Game Structures and GDL". In: *ECAI*. 2016, pp. 408–416.

[28] John McCarthy and Patrick J. Hayes. "Some Philosophical Problems from the Standpoint of Artificial Intelligence". In: *Machine Intelligence* 4 (1969), pp. 463–502.

[29] Raymond Reiter. *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. Cambridge, MA, USA: MIT Press, 2001. ISBN: 9780262182188.

[30] Hector J. Levesque, Fiora Pirri, and Raymond Reiter. "Foundations for the Situation Calculus". In: *Electronic Transactions on Artificial Intelligence (ETAI)* 2 (1998), pp. 159–178.

[31] Javier Pinto. "Concurrent Actions and Interacting Effects". In: *KR*. 1998, pp. 292–303.

[32] Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque. "ConGolog, A Concurrent Programming Language based on the Situation Calculus". In: *Artificial Intelligence* 121.1-2 (2000), pp. 109–169.

Chapter 5

Causal Responsibility Anticipation and Attribution in Situation Calculus Concurrent Game Structures

Abstract

Responsibility is a central concept in accountable decision making for multiagent systems. As modern AI systems grow in complexity and autonomy, there is a growing demand for them to address issues in AI ethics, prompting researchers to formalize responsibility from diverse perspectives, including strategic responsibility. However, causal responsibility, i.e. responsibility due to actual causal contribution, has received much less attention. In this paper, we study variants of responsibility attribution from both strategic and causal perspectives within a synchronous game-theoretic logic framework that allows concurrent moves by multiple agents. Our formalization

is based on Situation Calculus Synchronous Game Structures (SCSGS). We show that by combining these perspectives, one can obtain novel forms of responsibility attribution that are grounded on actual causation. While doing this, we propose an account of actual causation in SCSGS. We prove that our formalization handles the issues associated with preemption and over-determination well. We also study some key properties of responsibility and demonstrate that causal, strategic, and combined notions of responsibility are extensionally distinct.

5.1 Introduction

Responsibility is a central concept for accountable decision making in multiagent systems. As modern AI systems grow in complexity and autonomy, there is a growing demand for them to address issues in AI ethics, prompting researchers to formalize responsibility from diverse perspectives, including that of structural equation models [1], STIT logic [2, 3, 4, 5], ATL [6, 7], LTLf [8, 9], game-theory [10, 11, 12], and logics of strategic and extensive games [13, 14, 15]. Much of this research formalizes strategic responsibility [9], which involves assessing whether an agent's choice led to or "caused" a given outcome.[1] The literature distinguishes two main views on this. One, associated with Frankfurt [17], holds that an agent is responsible only if they could have acted otherwise. The other ties responsibility to making an outcome inevitable, i.e. a "seeing-to-it" view linked to STIT logic. Based on this, in their

[1]In the literature, this is sometimes called "causal responsibility" [16, 9], which may lead to confusion.

action-based framework, [3] proposed two forms of strategic responsibility, *active responsibility* and *passive responsibility*. The former pertains to an agent ensuring that some state of affairs occurs through their actions, while the latter involves the agent's failure to prevent that effect from occurring. [8] identified two variants of passive responsibility based on whether the reasoning occurs before or after the outcome. *Passive responsibility anticipation* is a future-looking or *ex ante* notion and involves determining whether a certain choice would incur some responsibility. *Passive responsibility attribution*, on the other hand, is a retrospective or *ex post* notion, which involves assigning responsibility after the choices have been made. Strategic notions of responsibility focus on the choices that agents have and whether they promote the outcome. In multiagent game settings, it is natural to see the outcome as being determined by the entire combination of agents' moves. In this case, even doing nothing is a choice that may lead to a particular outcome (e.g., a doctor's absence might result in a patient's death).

The problem of determining *Actual Causality* is the problem of identifying the causes of an observed effect from a given history of events or actions, which is also called the scenario [18]. For instance, in a scenario where a prison guard A loads a gun and prison guard B shoots an inmate with the gun, their actions may be identified as the cause of the inmate's death. In actual causality, one focuses on what effects the actions have. Doing nothing is not an actual cause.

The notions of responsibility and actual causality are closely related. Despite this, the connection between these remains largely unexplored. To deal with this,

in this paper we make a distinction between *passive causal responsibility attribution* (such as guard A's loading) and *passive strategic responsibility attribution* (such as the doctor's absence). Based on this, we propose notions of responsibility grounded in strategic reasoning as well as causal contribution. Our formalization is based on the Situation Calculus Synchronous Game Structures (SCSGS) [19], a game-theoretic logic framework that allows concurrent moves by multiple agents. We show that by combining causal and strategic perspectives, one can obtain novel and stronger forms of responsibility attribution that are extensionally distinct.

While doing this, we propose an account of actual causation in SCSGS. It overcomes a major limitation of previous proposals of actual causation in action-theoretic frameworks, which were in turn proposed to deal with the expressive limitations of structural equations models-based causal models [20]: that the scenario is a linear sequence of single-agent actions, and is thus restricted to turn-taking multiagent games. In contrast, we have a single action *tick* whose effects depend on the combination of moves selected by the players. Each agent selects its move without knowing which move is selected by the other agents. As we will see, in domains with synchronous concurrency, besides the usual preemption problem,[2] we also face the problem of overdetermination, as there may be more than one subset of the moves that are sufficient to cause the effect. In this paper, we extend previous accounts of actual causation in the situation calculus [21, 22] to identify minimal subsets of moves by some of the

[2]Preemption happens when two competing events try to achieve the same effect, and the latter of these fails to do so, as the earlier one has already achieved the effect.

agents that are causes of the effect, i.e., sufficient to cause it. We also identify causal chains consisting of such minimal sets of moves, and notice that there may be several of them in the scenario for a given effect.

Our contribution in this paper is thus fourfold: (i) We propose a formalization of actual causation in SCSGS. (ii) We extend previously proposed single-agent notions of strategic responsibility for coalitions of agents. (iii) Based on these, we formalize new notions of passive causal, strategic, and combined responsibility attribution. (iv) Finally, we prove some important properties of our formalization.

5.2 Preliminaries

Situation Calculus (SC). The SC is a well-known second-order language for representing and reasoning about dynamic worlds [23, 24]. In the SC, all changes are due to named actions, which are terms in the language. Situations represent a possible world history resulting from performing some actions. The constant S_0 is used to denote the initial situation where no action has been performed yet. The distinguished binary function symbol $do(a, s)$ denotes the successor situation to s resulting from performing the action a. The expression $do([a_1, \cdots, a_n], s)$ represents the situation resulting from executing actions $a_1, \cdots, a_n$, starting with situation s. As usual, a relational/functional fluent representing a property whose value may change from situation to situation takes a situation term as its last argument. There is a special predicate $Poss(a, s)$ used to state that action a is executable in situation s. Also,

the special binary predicate $s \sqsubset s'$ represents that s' can be reached from situation s by executing some sequence of actions. Moreover, $s \sqsubseteq s'$ is an abbreviation of $s \sqsubset s' \vee s = s'$. Again, $s < s'$ is an abbreviation of $s \sqsubset s' \wedge \mathit{Executable}(s')$, where $\mathit{Executable}(s)$ is defined as $\forall a', s'.\ \mathit{do}(a', s') \sqsubseteq s \supset \mathit{Poss}(a', s')$, i.e. every action performed in reaching situation s was possible in the situation in which it occurred. Finally, $s \leq s'$ means $s < s' \vee s = s'$.

In the SC, a dynamic domain is specified using a basic action theory (BAT) $\mathcal{D}$ that includes the following sets of axioms: (i) (first-order or FO) initial state axioms $\mathcal{D}_{S_0}$, which indicate what was true initially; (ii) (FO) action precondition axioms $\mathcal{D}_{ap}$, characterizing $\mathit{Poss}(a, s)$; (iii) (FO) successor-state axioms $\mathcal{D}_{ss}$, indicating precisely when the fluents change; (iv) (FO) unique-names axioms $\mathcal{D}_{una}$ for actions, stating that different action terms represent distinct actions; and (v) (second-order or SO) domain-independent foundational axioms Σ, describing the structure of situations [25]. Although the SC is SO, Reiter [24] showed that for certain type of queries ϕ, $\mathcal{D} \models \phi$ iff $\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \models \mathcal{R}[\phi]$, where $\mathcal{R}$ is a syntactic transformation operator called *regression* and $\mathcal{R}[\phi]$ is a SC formula that compiles dynamic aspects of the theory $\mathcal{D}$ into the query ϕ. Thus reasoning in the SC for a large class of interesting queries can be restricted to entailment checking w.r.t a FO theory [24].

Synchronous Game Structures (SCSGS). Following [19], we focus on games where there are n players/agents each of whom chooses a move at every time step. All such moves are executed *synchronously* and determine the next state of the game.

At each time step, the state of the game is fully observable by all agents, as are all past moves of every agent. To represent such multi-player synchronous games, we use a special class of BATs, called *situation calculus synchronous game structures (SCSGS)*, which are defined as follows.

Agents. A SCSGS $\mathcal{D}$ involves a finite set of n agents, and we use a subsort *Agents* of *Objects* which includes these finitely many agents $Ag_1, \ldots, Ag_n$, each denoted by a constant, and for which unique names $Ag_i \neq Ag_j$ for $i \neq j$ and domain closure $agent(x) \equiv x = Ag_1 \vee \cdots \vee x = Ag_n$ hold.

Moves. We also use a second subsort *Moves* of *Objects*, representing the possible moves. These come in finitely many types, represented by function symbols $M_i(\vec{x})$, which are parameterized by objects $\vec{x}$, with $Move(m) \equiv \bigvee_i \exists \vec{x}.m = M_i(\vec{x})$. Given that the parameters range over *Objects*, each agent may have an infinite number of possible moves at each time step. We have unique name and domain closure axioms (parameterized by objects) for these functions $M_i(\vec{x}) \neq M_j(\vec{y})$ for $i \neq j$, and $M_i(\vec{x}) = M_i(\vec{y}) \supset \vec{x} = \vec{y}$.

Actions. In SCSGS, there is only *one action type*, $tick(m_1, \ldots, m_n)$, which represents the execution of a joint move by all the agents at a given time step. The action *tick* has exactly n parameters, $m_1, \ldots, m_n$, one per agent, which are of sort *Moves* and corresponds to the simultaneous choice of the move to perform by the n different agents.

Legal moves. The *legal moves* available to each agent in a given situation are specified formally using a special predicate *LegalM*, which is defined by statements of the

following form (one for each agent Ag_i and move type M_i): $LegalM(Ag_i, M_i(\vec{x}), s) \stackrel{\text{def}}{=} \Phi_{Ag_i,M_i}(\vec{x}, s)$, i.e., agent Ag_i can legally perform move $M_i(\vec{x})$ in situation s if and only if $\Phi_{Ag_i,M_i}(\vec{x}, s)$ holds. Technically *LegalM* is an abbreviation for $\Phi_{Ag_i,M_i}(\vec{x}, s)$, which is a uniform formula (i.e., a formula that only refers to a single situation s).

Precondition axioms. The precondition axiom for the action *tick* is fixed and specified in terms of *LegalM* as follows: $Poss(tick(m_1, \ldots, m_n), s) \equiv \wedge_{i=1,\ldots,n} LegalM(Ag_i, m_i, s)$. Thus the joint action by all agents $tick(m_1, \ldots, m_n)$ is executable if and only if each selected move m_i is a legal move for agent Ag_i in situation s. Since we only have one action type *tick*, this is the only precondition axiom in $\mathcal{D}_{ap}$.

Successor state axioms. We have *successor state axioms* $\mathcal{D}_{ss}$, specifying the effects and frame conditions of the joint moves $tick(m_1, \ldots, m_n)$ on the fluents. Such axioms, as usual in basic action theories, are domain specific, and characterize the actual game under consideration. Within such axioms, the agent moves, which occur as parameters of *tick*, determine how fluents change as the result of joint moves.[3]

Initial situation description. Finally, the initial state of the game is axiomatized in the *initial situation description* $\mathcal{D}_{S_0}$ as usual, in a domain specific way.

SCSGS, as defined above, are a first-order extension of the Concurrent Game Structures used with logics such as ATL*, but they incorporate an action theory to specify how agent moves change the fluents and address the frame problem.

[3] In many cases, moves don't interfere with each other and the effects are just the union of those of each move. One can also exploit previous work on axiomatizing parallel actions to generate successor state axioms [24, 26].

LTL Properties. In formalizing strategic responsibility, we will use LTL temporal properties. To do this, we utilize the axiomatization of infinite paths in the SC introduced by [27], which adds a new sort of paths to the language that provides a natural way to talk about "infinite future histories". Paths are infinite sequences of executable situations. Following this work, we will use the special predicates $OnPath(p, s)$ to mean that situation s is on path p, $Starts(p, s)$ to specify that the path p starts with situation s, and $Suffix(p', p, s)$ to denote that the path p' starts with s and contains the same situations as p starting from s. Based on this, [28] defined a special predicate $Holds(\phi, p)$ to specify that a given LTL property ϕ holds on path p. In the following, we use ϕ to range over LTL formulae.

Bottle Example. We use a variant of the well-known "bottle" example [29], where Suzy and Billy are throwing stones at a bottle. Suzy's stones are smaller and thus she requires two throws to break the bottle while Billy's stone is large and he needs just one throw to break it. The available moves of $ag \in \{Suzy, Billy\}$ can be one of $pick_{ag}$, representing the picking of stone(s), one for $ag = Billy$ or two for $ag = Suzy$; $throw_{ag}$, i.e. throwing of a stone by ag; and a catchall $other_{ag}$ move, denoting anything other than picking and throwing. The legality of these moves is specified below. (a). $LegalM(pick_{ag}, s) \stackrel{\text{def}}{=} \neg Holding(ag, s)$. (b).$LegalM(throw_{ag}, s) \stackrel{\text{def}}{=} Holding(ag,\ s)$. (c). $LegalM(other_{ag}, s)$. Thus, e.g., throwing a stone is a legal move for agent ag in situation s is she is holding one or more stones in s. For simplicity, we assume that the $other_{ag}$ move is always possible.

There are three fluents in this domain, $Holding(ag, s)$, $Broken(s)$, and $SuzyThrown(s)$, which means that the agent ag is holding their stones in situation s, the bottle is broken in s, and $Suzy$ has already thrown once before in s, respectively. The successor-state axioms are as follows.

$$
\begin{aligned}
&(d).\ Holding(ag, do(a, s)) \equiv \\
&\quad [ag = Suzy \wedge \exists m.\ a = tick(pick_{Suzy}, m)] \vee \\
&\quad [ag = Billy \wedge \exists m.\ a = tick(m, pick_{Billy})] \vee \\
&\quad [ag = Suzy \wedge Holding(ag, s) \wedge \\
&\qquad \neg(\exists m.\ a = tick(throw_{Suzy}, m) \wedge SuzyThrown(s))] \vee \\
&\quad [ag = Billy \wedge Holding(ag, s) \wedge \\
&\qquad \neg\exists m.\ a = tick(m, throw_{Billy})],
\end{aligned}
$$

$$
\begin{aligned}
&(e).\ Broken(do(a, s)) \equiv [\exists m.\ a = tick(m, throw_{Billy})] \vee \\
&[\exists m.\ a = tick(throw_{Suzy}, m) \wedge SuzyThrown(s)] \vee Broken(s),
\end{aligned}
$$

$$
\begin{aligned}
&(f).\ SuzyThrown(do(a, s)) \equiv \\
&\quad \exists m.\ a = tick(throw_{Suzy}, m) \vee SuzyThrown(s).
\end{aligned}
$$

Finally, we specify what is true initially in S_0: $(g).\ \forall ag.\ \neg Holding(ag, S_0)$. $(h).\ \neg Broken(S_0)$. We will use $\mathcal{D}_{bt}$ to refer to this axiomatization. □

5.3 Actual Causation in the SC

Based on Batusov and Soutchanski's [21] original proposal, Khan and Lespérance [22] (KL) recently defined cause in the SC. Both assume that the scenario is a linear sequence of actions, i.e. these do not allow concurrent actions.

KL introduced the notion of *dynamic formulae*. An effect φ in their framework is a situation-suppressed dynamic formula.[4] Given an effect φ, the actual causes are defined relative to a scenario s. When s is ground, the tuple $\langle \varphi, s \rangle$ is called a *causal setting*. Also, it is assumed that $\mathcal{D} \models \mathit{Executable}(s) \wedge \neg\varphi[S_0] \wedge \varphi[s]$. Here $\varphi[s]$ denotes the formula obtained from φ by restoring the appropriate situation argument into all fluents in φ (see Def. 4.3.2).

KL required that each situation be associated with a time-stamp, which can then be used to uniquely identify an action occurrence. A time-stamp is an integer for their theory. KL assumed that the initial situation starts at time-stamp 0 and each action increments the time-stamp by one. Thus, their action theory includes the following axioms: $\mathit{timeStamp}(S_0) = 0$; $\forall a, s, ts.\ \mathit{timeStamp}(do(a, s)) = ts \equiv \mathit{timeStamp}(s) = ts - 1$. With this, causes in their framework is a non-empty set of action-time-stamp pairs.

The notion of *dynamic formulae* is defined as follows:

Definition 5.3.1. *Let $\vec{x}$, θ_a, and $\vec{y}$ respectively range over object terms, action terms,*

[4] While KL also study epistemic causation, we restrict our discussion to objective causality only. Also, in the following, we will use the terms situation, scenario, and history interchangeably.

and object and action variables. The class of dynamic formulae φ *is defined inductively using the following grammar:* $\varphi ::= P(\vec{x}) \mid Poss(\theta_a) \mid After(\theta_a, \varphi) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists\vec{y}.\ \varphi$.

We will use φ for DF. $\varphi[\cdot]$ is defined as follows:

Definition 5.3.2.

$$\varphi[s] \stackrel{\text{def}}{=} \begin{cases} P(\vec{x}, s) & \text{if } \varphi \text{ is } P(\vec{x}) \\ Poss(\theta_a, s) & \text{if } \varphi \text{ is } Poss(\theta_a) \\ \varphi'[do(\theta_a, s)] & \text{if } \varphi \text{ is } After(\theta_a, \varphi') \\ \neg(\varphi'[s]) & \text{if } \varphi \text{ is } (\neg\varphi') \\ \varphi_1[s] \wedge \varphi_2[s] & \text{if } \varphi \text{ is } (\varphi_1 \wedge \varphi_2) \\ \exists\vec{y}.\ (\varphi'[s]) & \text{if } \varphi \text{ is } (\exists\vec{y}.\ \varphi') \end{cases}$$

Recently, [30] proposed a variant of KL's definition of cause that captures the causal chain while identifying causes. Here we briefly discuss this. The idea behind how causes are identified is as follows. Given an effect φ and scenario s, if some action of the action sequence in s triggers φ to change its truth value from false to true relative to $\mathcal{D}$, and if there are no actions in s after it that change the value of φ back to false, then this action is a *primary* or *direct* actual cause of achieving φ in s, denoted using $CausesDirectly(a, ts, \varphi, s)$.

Definition 5.3.3 (Primary Cause (KL 2021))**.**

$$\mathit{CausesDirectly}(a, ts, \varphi, s) \stackrel{\text{def}}{=} \exists s_a.\ \mathit{timeStamp}(s_a) = ts \wedge$$

$$S_0 < do(a, s_a) \leq s \wedge \neg\varphi[s_a] \wedge \forall s'.(do(a, s_a) \leq s' \leq s \supset \varphi[s']).$$

That is, a executed at time-stamp ts is the *primary cause* of effect φ in situation s iff a was executed in a situation with time-stamp ts in scenario s, a caused φ to change its truth value to true, and no subsequent actions on the way to s falsified φ.

Now, note that a (primary) cause a might have been non-executable initially. Also, a might have only brought about the effect conditionally and this context condition might have been false initially. Thus earlier actions in the trace that contributed to the preconditions and the context conditions of a cause must be considered as causes as well. $\mathit{CausesByChain}(a, ts, cc, \varphi, s)$, which means that action a at timestamp ts is a cause of an effect φ in scenario s through causal chain cc, inductively captures all such primary and indirect causes and specifies the causal chain.[5]

[5]In this, we need to quantify over situation-suppressed DF. Thus we must encode such formulae as terms and formalize their relationship to the associated SC formulae. This is tedious but can be done essentially along the lines of [31]. We assume that we have such an encoding and use formulae as terms directly.

Definition 5.3.4 (Actual Cause Through Causal Chain)**.**

$$
\begin{aligned}
&\mathit{CausesByChain}(a, ts, cc, \varphi, s) \stackrel{\text{def}}{=} \\
&\forall P.[\forall a, ts, s, cc, \varphi.(\mathit{CausesDirectly}(a, ts, \varphi, s) \\
&\qquad \supset P(a, ts, ((a, ts)), \varphi, s)) \\
&\quad \wedge \forall a, ts, cc', s, \varphi.(\exists a', ts', s'.(\mathit{CausesDirectly}(a', ts', \varphi, s) \\
&\qquad \wedge \mathit{timeStamp}(s') = ts' \wedge s' < s \\
&\qquad \wedge P(a, ts, cc', [\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)], s') \\
&\qquad \wedge cc = \mathit{Append}(cc', (a', ts')) \\
&\qquad \supset P(a, ts, cc, \varphi, s)) \\
&\quad] \supset P(a, ts, cc, \varphi, s).
\end{aligned}
$$

Thus, *CausesByChain* is defined to be the least relation P such that if a executed at time-stamp ts directly causes φ in scenario s then (a, ts, cc, φ, s) is in P, where $cc = ((a, ts))$; and if a' executed at ts' is a direct cause of φ in s, the time-stamp of s' is ts', $s' < s$, and $(a, ts, cc', [\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)], s')$ is in P (i.e. a executed at ts is a direct or indirect cause of $[\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)]$ in s' through causal chain cc'), then (a, ts, cc, φ, s) is in P, where $cc = \mathit{Append}(cc', (a', ts'))$. Here the effect $[\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)]$ requires a' to be executable and φ to hold after a'. Also, *Append* is defined as follows: $\mathit{Append}(((a_1, ts_1), \ldots, (a_n, ts_n)), (a, ts)) \stackrel{\text{def}}{=} ((a_1, ts_1), \ldots, (a_n, ts_n), (a, ts))$.

Tick Actions as Causes in the SCSGS. The above formalization of actual causation was formulated for domains specified by BATs in the situation calculus. However, it can be used directly for SCSGS domains, as long as one focuses on identifying the *tick* actions in the scenario that caused the effect, and causal chains consisting of *tick* actions. This is not surprising as SCSGS are special kinds of BATs. We illustrate this in the example below.

Example (cont'd). Consider the scenario $\sigma_1 = do([tick(pick_{Suzy}, other_{Billy}), tick(throw_{Suzy}, pick_{Billy}), tick(other_{Suzy}, other_{Billy}), tick(throw_{Suzy}, throw_{Billy})], S_0)$. We want to find the actual causes of the effect $\varphi_1 = Broken(s)$. We can show that:[6] $\mathcal{D}_{bt} \models CausesByChain(tick(pick_{Suzy}, other_{Billy}), 0, cc, \varphi_1, \sigma_1)$, where $cc = ((tick(pick_{Suzy}, other_{Billy}), 0), (tick(thr- ow_{Suzy}, pick_{Billy}), 1), (tick(throw_{Suzy}, throw_{Billy}), 3))$. Explaining backward in *cc*, the last *tick* action executed at time-stamp 3 is included in the causal chain as (either of the moves in) it directly caused the breaking of the bottle. The second *tick* action executed at 1 is also included because it is a (secondary/indirect) cause as it brought about the preconditions of the last *tick* action (by making Billy's throw legal), besides bringing about the context condition (that *SuzyThrown*) under which Suzy's second throw can break the bottle. Finally, the first *tick* action is also a cause as it made the second *tick* action executable. □

While the above formalization provides some insight on what *tick* actions are

[6]Note that since the definition of *CausesByChain* inductively constructs the causal chain, considering some of the causes, e.g., the primary cause, will only give us a suffix of the complete causal chain *cc*; for simplicity, we thus only show the most indirect cause below, which captures the complete chain *cc*.

causes and can be used to identify the completely irrelevant *tick* actions, e.g. the one at time-stamp 2, observe that some irrelevant moves might still be included in the discovered causes, such as $other_{Billy}$ at time-stamp 0 in our example. In other words, our formalization of this does not specify what moves within the identified *tick* actions are contributing to the effect. To deal with this, we next propose a formalization of agent moves as causes.

5.4 Agent Moves as Causes in the SCSGS

We now go a step further by pinpointing the moves that actually contributed to the effect within the *tick* actions that are identified as causes. Note that, since unlike actions, agent moves within each *tick* action are concurrently performed, it is possible that more than one alternative chain of subsets of moves in the scenario are each by itself sufficient to bring about the effect. For instance, in our example, either $((pick_{Suzy}, 0), (throw_{Suzy}, 1), (throw_{Suzy}, 3))$ or $((pick_{Billy}, 1), (throw_{Billy}, 3))$ would have been sufficient to break the bottle. Just as with causal chains in the SC, we will identify these refined causal chains in two steps. In the first step, we identify the minimal set of moves in each action that is a direct cause of the effect in some refined chain (e.g., $throw_{Billy}$ in the last *tick* of the second refined causal chain above). We call these sets of direct causes *minimal moves primary causes* since they are minimal sets of moves that are causes. However, we must consider that there might be more than one minimal moves primary cause in one single action. For example in the last

tick of our example, we can consider either only Suzy's throwing or Billy's throwing to be a minimal moves primary cause. In the second step, using refined causes, we define the refined chains mentioned above, which we call *minimal moves causal chains*.

In keeping with the formalization of dynamic domains in the SC, we will consider actions (but not moves) as (refined) causes.[7] Thus our formalization of this does not omit the irrelevant moves in each time-stamp altogether, but rather replaces them with the special move *wait* within the *tick* action to remove their effects. *wait* has no effects (the domain modeler must ensure this), and is always legal: $\forall ag, s.\ LegalM(ag, wait, s)$.[8] Thus, for instance, in our example, one such refined action that is a cause is $tick(pick_{Suzy}, wait)$. We collect all of these for all causal chains in our new definition of causes.

We now define minimal moves primary causes:

Definition 5.4.1 (Minimal Moves Primary Cause)**.**

$$MinMovCausesDir(a, ts, (a', ts), \varphi, s) \stackrel{\text{def}}{=}$$

$$\exists s_a.\ timeStamp(s_a) = ts \wedge (S_0 < do(a, s_a) \leq s) \wedge \neg\varphi[s_a] \wedge$$

$$\forall s'.(do(a, s_a) \leq s' \leq s \supset \varphi[s']) \wedge MinSuffSubset(a', a, s_a, \varphi, s).$$

The above definition is exactly as the definition of primary causes (Def. 5.3.3), but has

[7] Note that the action theory specifies how the situation changes when actions, i.e., joint moves, are performed. This allows interfering or synergic effects to be specified.

[8] In many settings, an agent might be forced to make a non-*wait* move, e.g. in chess. But our simplifying assumption that the agent can always *wait* is only used to extract the contributions of her moves for the purpose of causation. We could add constraints to rule out such moves in real play.

an additional parameter (a', ts) that returns a tuple consisting of a minimal subset of moves a' of the primary cause a that, when executed in s_a (i.e., the situation where the primary cause was executed in the original scenario), is sufficient to cause the effect φ in s (formalized using *MinSuffSubset*), and the time stamp ts of a. $MinSuffSubset(a', a, s', \varphi, s)$ means that the *tick* action a' consists of a sufficient subset of moves of a in s' to achieve φ up to s, and it is minimal.

Definition 5.4.2.

$$MinSuffSubset(a', a, s', \varphi, s) \stackrel{\text{def}}{=} SuffSubset(a', a, s', \varphi, s)$$
$$\wedge \neg\exists a^*.\ a^* \neq a' \wedge SuffSubset(a^*, a', s', \varphi, s).$$

a' is a sufficient subset of moves of a in s' to achieve φ up to s, i.e. $SuffSubset(a', a, s', \varphi, s)$, iff a' is a subset of moves of a, and the execution of this subset a' in s' is sufficient to cause φ in s, i.e. φ becomes true after a' is executed in s' and it remains true after the subsequent actions between $do(a, s')$ and s are executed starting in $do(a', s')$.

Definition 5.4.3.

$$
\begin{aligned}
&\mathit{SuffSubset}(a', a, s', \varphi, s) \stackrel{\text{def}}{=} \\
&\mathit{SubsetMovs}(a', a) \wedge \exists s''.\ \mathit{timeStamp}(s'') = \mathit{timeStamp}(s) \\
&\quad \wedge s' < s'' \wedge \forall a_1, s_1, a_1', s_1'.[\\
&\qquad (\mathit{do}(a, s') < \mathit{do}(a_1, s_1) \leq s \wedge \\
&\qquad\ \mathit{do}(a', s') < \mathit{do}(a_1', s_1') \leq s'' \wedge \\
&\qquad\ \mathit{timeStamp}(s_1') = \mathit{timeStamp}(s_1)) \supset (a_1 = a_1')] \\
&\quad \wedge (\forall s_1'.(s' < s_1' \leq s'') \supset \varphi[s_1']).
\end{aligned}
$$

Here $\mathit{SubsetMovs}(a', a)$, meaning that the *tick* action a' is exactly as a, but with some of the moves possibly replaced with the *wait* move, is defined as follows:

Definition 5.4.4 (a' Consists of a Subset of a)**.**

$$
\begin{aligned}
&\mathit{SubsetMovs}(a', a) \stackrel{\text{def}}{=} \\
&\exists m_1', \cdots, m_n', m_1, \cdots, m_n.\ a' = \mathit{tick}(m_1', \cdots, m_n') \\
&\qquad \wedge a = \mathit{tick}(m_1, \cdots, m_n) \\
&\qquad \wedge (\forall j.\ 1 \leq j \leq n \supset (m_j' = m_j \vee m_j' = \mathit{wait})).
\end{aligned}
$$

Using minimal moves primary causes, we next formalize minimal moves causal chains, a variant of *CausesByChain* that inductively constructs the minimal moves chain instead.

Definition 5.4.5 (Min. Moves Cause by Causal Chain)**.**

$$
\begin{aligned}
&\mathit{MinMovesCausesByChain}(a, ts, cc, \varphi, s) \stackrel{\text{def}}{=} \\
&\forall P.[\forall a, ts, cc, a', s, \varphi.[\mathit{MinMovCausesDir}(a, ts, (a', ts), \varphi, s) \\
&\qquad\qquad \supset\ P(a, ts, ((a', ts)), \varphi, s)] \\
&\quad \wedge \forall a, ts, cc', s, \varphi.[\exists a'', a', ts', s'.(\\
&\qquad \mathit{MinMovCausesDir}(a', ts', (a'', ts'), \varphi, s) \\
&\qquad\quad \wedge\ \mathit{timeStamp}(s') = ts' \wedge s' < s \\
&\qquad\quad \wedge\ P(a, ts, cc', [\mathit{Poss}(a'') \wedge \mathit{After}(a'', \varphi)], s') \\
&\qquad\quad \wedge\ cc = \mathit{Append}(cc', (a'', ts'))) \supset\ P(a, ts, cc, \varphi, s)] \\
&\quad] \supset P(a, ts, cc, \varphi, s).
\end{aligned}
$$

Thus *MinMovesCausesByChain* is the smallest set such that if a *tick* action a executed at time-stamp ts directly caused the effect φ in scenario s with the minimal moves primary cause (a', ts), then (a, ts, cc, φ, s) is in that set, where $cc = ((a', ts))$; and if a' executed at ts' is a direct cause of φ in s with minimal moves primary cause (a'', ts'), the time-stamp of s' is ts', $s' < s$, and $(a, ts, cc', [\mathit{Poss}(a'') \wedge \mathit{After}(a'', \varphi)], s')$ is in P (i.e. a executed at ts is a direct or indirect cause of $[\mathit{Poss}(a'') \wedge \mathit{After}(a'', \varphi)]$ in s' through minimal moves causal chain cc'), then (a, ts, cc, φ, s) is in P, where $cc = \mathit{Append}(cc', (a'', ts'))$. This thus incrementally constructs the refined causal chains using *MinMovCausesDir*.

Minimal moves causal chains, as defined above, can be incomplete and might only

include part of the chain. Thus, we also define a notion of complete chains.

Definition 5.4.6 (Complete Min. Moves Causal Chain)**.**

$$\begin{aligned}
&\mathit{CompleteMinMovesCausalChain}(cc, \varphi, s) \stackrel{\text{def}}{=} \\
&\exists a, ts.\ \mathit{MinMovesCausesByChain}(a, ts, cc, \varphi, s) \wedge \\
&\forall cc', ts', a'. cc' \neq cc \wedge \mathit{MinMovesCausesByChain}(a', ts', cc', \varphi, s) \\
&\qquad \supset (\exists a^*, ts^*.\ (a^*, ts^*) \in cc \wedge (a^*, ts^*) \notin cc').
\end{aligned}$$

Thus cc is a complete minimal moves causal chain given effect φ and scenario s iff cc is a minimal moves causal chain for some *tick* action a and timestamp ts, and cc includes a minimal moves cause (a^*, ts^*) that no other distinct minimal moves causal chains cc' of φ and s includes. As shown below, there can be more than one complete minimal moves causal chains.

Example (cont'd). We can show the following result.

Proposition 5 (Complete Causal Chains in σ_1)**.**

$$\begin{aligned}
&\mathcal{D}_{bt} \models \mathit{CompleteMinMovesCausalChain}(cc_1, \varphi_1, \sigma_1) \wedge \\
&\qquad \mathit{CompleteMinMovesCausalChain}(cc_2, \varphi_1, \sigma_1), \\
&\text{where } cc_1 = ((\mathit{tick}(\mathit{pick}_{Suzy}, \mathit{wait}),0),(\mathit{tick}(\mathit{throw}_{Suzy}, \mathit{wait}),1), \\
&(\mathit{tick}(\mathit{throw}_{Suzy}, \mathit{wait}), 3)), \text{ and } cc_2 = ((\mathit{tick}(\mathit{wait}, \mathit{pick}_{Billy}),1), \\
&(\mathit{tick}(\mathit{wait}, \mathit{throw}_{Billy}),3)).
\end{aligned}$$

Thus, in our example, we have two distinct complete causal chains, one that stems from the refined primary cause involving Suzy's second throw move and Billy's *wait* move at time-stamp 3, and another originating from Billy's throw and Suzy's *wait* move, again at time-stamp 3. □

5.5 Properties of Actual Causation in SCSGS

Preemption. Preemption occurs when there are more than one competing contributors (actions) to an effect, but they happen one after another/consecutively. In such cases, only the first of these should be identified as the actual cause. The (effects of the) latter actions are said to be preempted by the actual cause. We illustrate this below.

Example 2. Consider the new scenario $\sigma_2 = do([tick(pick_{Suzy}, pick_{Billy}),$ $tick(throw_{Suzy}, other_{Billy}), tick(throw_{Suzy}, other_{Billy}), tick(other_{Suzy}, throw_{Billy})], S_0)$. In this, we can show the following result.

Proposition 6.

$$\mathcal{D}_{bt} \models \neg\exists cc.\ CausesByChain(tick(other_{Suzy}, throw_{Billy}), 3,$$
$$cc, \varphi_1, \sigma_2) \wedge \neg\exists cc, m.\ MinMovesCausesByChain(tick(m,$$
$$throw_{Billy}), 3, cc, \varphi_1, \sigma_2).$$

Thus as expected, Billy's throw is not considered as part of any (refined) causal

chain. □

Over-determination. Over-determination happens when the effect is contributed by some events, but a smaller subset of these would have been sufficient for the effect to hold. For example, in a voting scenario, where 6 out of 10 votes are required for a candidate to win, if 7 voters voted for candidate A, saying that all of these 7 votes are the cause of candidate A's winning the ballot would be over-determination as any 6 of these would suffice. An acceptable solution to this is to only identify any 6-vote subsets as causes [20].

Example (cont'd). In our first bottle example, there are only two refined causal chains; the first one only consists of Billy's moves, and the second only consists of Suzy's. The definition *MinMovesCausesByChain* ensures that only these two chains exist, since if we were to consider the chain that includes the time-stamp 3 throw moves of both Suzy and Billy, it will not be a minimal. Formally:

Proposition 7.

$$\mathcal{D}_{bt} \models \neg\exists a, ts, cc.\ MinMovesCausesByChain(a, ts,$$
$$Append(cc, (tick(throw_{Suzy}, throw_{Billy}), 3)), \varphi_1, \sigma_1).$$

Thus, $tick(throw_{Suzy}, throw_{Billy})$ at time 3 cannot be a part of any refined causal chain for any action and timestamp. □

In general, since refined causal chains are constructed to be minimal, if we have

two refined chains in the same timestamp, it cannot be the case that one of them is a refined version of the other (i.e., has a subset of moves of the other).

Theorem 5.5.1 (No Over-Determination)**.**

$$\begin{aligned}
&\mathcal{D} \models \forall a, a_1, a_2, ts, cc, \varphi, s. \\
&(\mathit{MinMovesCausesByChain}(a, ts, \mathit{Cons}((a_1, ts), cc), \varphi, s) \\
&\wedge \mathit{MinMovesCausesByChain}(a, ts, \mathit{Cons}((a_2, ts), cc), \varphi, s)) \\
&\supset (a_1 = a_2 \vee (\neg \mathit{SubsetMovs}(a_1, a_2) \wedge \neg \mathit{SubsetMovs}(a_2, a_1))).
\end{aligned}$$

Here $\mathit{Cons}((a, ts), cc)$ denotes the sequence where (a, ts) is added to the front of cc.

Proof Sketch: By induction on the length of the minimal moves causal chain using properties of minimal sufficient subsets of joint moves. □

Persistence. Finally, we study the conditions under which (refined) causal chains persist when the scenario changes.

Theorem 5.5.2 (Persistence of the Causal Chain)**.**

$$\begin{aligned}
\mathcal{D} \models \forall s, s', cc, ts, a, \varphi.\ &\mathit{CausesByChain}(a, ts, cc, \varphi, s) \\
&\wedge s \leq s' \wedge \forall s^*, a^*.(s \leq do(a^*, s^*) \leq s' \supset \varphi[s^*]) \\
&\supset \mathit{CausesByChain}(a, ts, cc, \varphi, s').
\end{aligned}$$

Proof Sketch: By induction on the length of the causal chain. □

That is, if a *tick* action a executed in ts is the cause of an effect φ in scenario

s through causal chain cc, then a in ts remains the cause of φ in all subsequent situations/scenarios s' if φ does not change after it was achieved in s. This is because since the situation where φ was achieved does not change in the extended scenario, neither does the causal chain.

A similar result can be shown for refined causal chains.

Theorem 5.5.3 (Persistence of Refined Causal Chains)**.**

$$\mathcal{D} \models \forall s,s',cc,ts,a,\varphi.\ \mathit{MinMovesCausesByChain}(a, ts, cc, \varphi, s)$$
$$\wedge\, s \leq s' \wedge \forall s^*, a^*.(s \leq do(a^*, s^*) \leq s' \supset \varphi[s^*])$$
$$\supset \mathit{MinMovesCausesByChain}(a, ts, cc, \varphi, s').$$

Proof Sketch: By induction on the length of the minimal moves causal chain using properties of minimal sufficient subsets of joint moves. □

5.6 Causal, Strategic, Combined Responsibility

Before we can give our formalization of responsibility in SCSGS, we need to define some notions to support strategic reasoning in SCSGS. We define an agent strategy f_{ag} as a function from situations to agent ag's move in that situation, i.e. $f_{ag}(s) = m_{ag}(\vec{x})$. For a coalition of agents C, a joint strategy $\vec{f_C} = \cup_{ag \in C} f_{ag}$ and $\vec{f}_{C,ag}$ is $f_{ag} \in \vec{f_C}$.

[28] defined a notion of an agent being able to force an LTL temporal property ϕ by following a strategy f when operating in a nondeterministic domain, where the environment determines the outcome of the agent's actions. Here, we adapt this for

multiagent settings modelled as SCSGS:

Definition 5.6.1 (Multi-Agent CanForceBy)**.**

$$\mathit{CanForceBy}(C, \phi, \vec{f}_C, s) \stackrel{\text{def}}{=} \forall p. \mathit{Out}(p, C, \vec{f}_C, s) \supset \mathit{Holds}(\phi, p),$$

$$\mathit{where,}\ \mathit{Out}(p, C, \vec{f}_C, s) \stackrel{\text{def}}{=} \mathit{Starts}(p, s) \wedge$$

$$\forall a, s'. \mathit{OnPath}(p, s') \wedge \mathit{OnPath}(p, \mathit{do}(a, s')) \supset$$

$$\forall ag. (ag \in C \supset \mathit{AgentMove}(a, ag) = \vec{f}_{C,ag}(s')),$$

$\mathit{and,}\ \mathit{AgentMove}(\mathit{tick}(m_1, ..., m_i, ..., m_N), i) = m_i.$

That is, coalition C can force temporal property ϕ using strategy $\vec{f}_C$ starting in situation s against any possible moves by agents outside C iff ϕ holds over all paths that can result from C following $\vec{f}_C$ starting in s. Here, $\mathit{Out}(p, C, \vec{f}_C, s)$ means that path p is a possible outcome trace when C executes strategy $\vec{f}_C$ starting from situation s.

We also define a variant of the above that identifies the minimal set of agents that can force an effect.

Definition 5.6.2 (Minimal Multi-Agent CanForceBy)**.**

$$\mathit{MinCanForceBy}(C, \phi, \vec{f}_C, s) \stackrel{\text{def}}{=} \mathit{CanForceBy}(C, \phi, \vec{f}_C, s)$$

$$\wedge \neg \exists C', \vec{f}_{C'}.\ C' \subset C \wedge \mathit{CanForceBy}(C', \phi, \vec{f}_{C'}, s).$$

Note that, for a given ϕ and s, *MinCanForceBy* does not necessarily hold for a

unique C.

Finally, we define a predicate stating that situation s is consistent with C following strategy $\vec{f}_C$ starting from s':

Definition 5.6.3 (Strategy $\vec{f}_C$ is Consistent with Sit. s)**.**

$$\begin{aligned} &\mathit{ConsStrategySit}(C, s, \vec{f}_C, s') \stackrel{\text{def}}{=} \\ &\forall a^*, s^*, ag.\ s' < do(a^*, s^*) \leq s \wedge ag \in C \supset \\ &\vec{f}_{C,ag}(s^*) = \mathit{AgentMove}(a^*, ag). \end{aligned}$$

Strategic Responsibility. We are now ready to define various notions of strategic responsibility. For this, we closely follow the definitions for the single agent case presented in [8, 9], but extend these for a coalition C and SCSGS. In what follows, we use path formulae in the context of *CanForceBy* and dynamic formulae in that of causal chains.

We start with active responsibility. According to [8], an agent is actively responsible for an outcome under some strategy if (i) the strategy she selected forces that outcome, and (ii) it was possible for her to select an alternative strategy that would have not forced that outcome for at least some environment response. We extend this idea for a coalition of agents C with a strategy $\vec{f}_C$.

Definition 5.6.4 (Active Responsibility By)**.**

$$\mathit{ActiveRespBy}(C, \vec{f}_C, \varphi, s) \stackrel{\text{def}}{=} \mathit{MinCanForceBy}(C, \Diamond\varphi, \vec{f}_C, s)$$

$$\wedge\, \exists \vec{g}_C, \vec{g'}_{\overline{C}}.\ \mathit{CanForceBy}(C \cup \overline{C}, \Box\neg\varphi, (\vec{g}_C, \vec{g'}_{\overline{C}}), s),$$

where $(\vec{g}_C, \vec{g'}_{\overline{C}})$ *represents the strategy for each agent* i *such that for any situation* s, $(\vec{g}_C, \vec{g'}_{\overline{C}})(s) = g_i(s)$ *if* $i \in C$, *and* $(\vec{g}_C, \vec{g'}_{\overline{C}})(s) = g'_i(s)$, *otherwise.*

Thus, a coalition C is actively responsible for φ using strategy $\vec{f}_C$ starting in situation s iff it can force φ to eventually hold using $\vec{f}_C$ starting in s, and there is another strategy $\vec{g}_C$ for C that could have been used to always avoid φ starting in s at least for some strategy $\vec{g'}_{\overline{C}}$ of the rest of the agents $\overline{C}$.

Next, we define passive responsibility anticipation. According to [9], an agent anticipates (weak) passive responsibility for an outcome under some strategy F if (i) there exists an environment strategy G such that F and G together brings about the outcome, and (ii) there exists an agent strategy F' such that F' along with the same environment strategy (i.e. G) would not bring about φ. Again, we extend this idea for a coalition of agents C:

Definition 5.6.5 (Passive Responsibility Anticipation By)**.**

$$\mathit{PassiveRespAntBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s) \stackrel{\text{def}}{=}$$

$$\mathit{CanForceBy}(C \cup \overline{C}, \Diamond\varphi, (\vec{f}_C, \vec{f}_{\overline{C}}), s) \wedge$$

$$\exists \vec{g}_C.\, \mathit{CanForceBy}(C \cup \overline{C}, \Box\neg\varphi, (\vec{g}_C, \vec{f}_{\overline{C}}), s),$$

where $(\vec{f}_C, \vec{f}_{\overline{C}})$ *and* $(\vec{g}_C, \vec{f}_{\overline{C}})$ *is as defined above in Def. 5.6.4.*

Thus, C passively anticipates responsibility for φ using strategies $\vec{f}_C$ and $\vec{f}_{\overline{C}}$ starting in s iff C can force φ to eventually hold using $\vec{f}_C$ starting in s if the other agents followed $\vec{f}_{\overline{C}}$, and C also has a strategy $\vec{g}_C$ to ensure that φ always remains false starting in s if the other agents followed $\vec{f}_{\overline{C}}$.

Finally, we define a retrospective notion of passive responsibility. According to [9], an agent has (weak) passive responsibility for some outcome under strategy F and history H such that the outcome holds in H, if (i) there exists an environment strategy G such that H is consistent with F and G, and (ii) there exists another agent strategy F' such that when executed along with G, it would have brought about the opposite outcome. We now generalize this for a coalition C:

Definition 5.6.6 (Passive Responsibility Attribution By)**.**

$$\begin{aligned}
&\mathit{PassiveRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s) \stackrel{\text{def}}{=} \\
&\quad \varphi[s] \wedge \mathit{ConsStrategySit}(C, s, \vec{f}_C, S_0) \wedge \\
&\quad \mathit{ConsStrategySit}(\overline{C}, s, \vec{f}_{\overline{C}}, S_0) \wedge \exists s'.s' < s \wedge \\
&\quad \mathit{PassiveRespAntBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s').
\end{aligned}$$

Thus, coalition C has passive responsibility for outcome φ in situation/history s by strategies $\vec{f}_C$ and $\vec{f}_{\overline{C}}$ iff φ is observed in s, s is consistent with both $\vec{f}_C$ and $\vec{f}_{\overline{C}}$ starting in the initial situation S_0, and C could have anticipated passive responsibility

for φ by $\vec{f}_C$ and $\vec{f}_{\overline{C}}$ in an earlier situation s' in the history of s.[9]

Attempted Murder Example. Consider a domain $\mathcal{D}_{AM}$, in which there are four agents, two killers *kllr1* and *kllr2*, a *waiter*, and a *guard*. The killers each can *poison* a drink with 50% lethal strength, and can *serve* it to a victim. The waiter can also *serve* the drink. The guard can *intervene* if the drink is poisoned, before it is served (or at the same time it is being served). All agents can also instead choose to play *other* moves. The outcome is $\mathit{MurderAttempted}(s)$, which becomes true if the drink is 100% poisoned and then served without the guard intervening.

Now, consider the following individual strategies f_{ag} of agent $ag \in \{\mathit{kllr1}, \mathit{kllr2}, \mathit{waiter}, \mathit{guard}\}$. Agent *kllr1* poisons in the first tick and plays the 'other' move in the second and third. Agent *kllr2* poisons in the first tick, plays other in the second, and serves the drink in the third if it is not served yet, otherwise plays other in the third tick as well. Agent *guard* simply plays other in all three ticks. Finally, agent *waiter* serves in the second tick and chooses to play other in the first tick and in the third tick. In the following, we will combine these individual strategies f_{ag} to obtain strategies $\vec{f}_C$ for various coalitions C or their complements $\overline{C}$. Thus, e.g., if $C_1 = \{\mathit{kllr1}, \mathit{kllr2}\}$, we will use $\vec{f}_{C_1}$ to denote the strategy of this coalition where the individual strategies of each member ag of the coalition (in this case, *kllr1* and *kllr2*)

[9]Note that unlike in the case for active responsibility, our definitions of passive responsibility do not ensure that the responsible group is minimal. With some effort, passive responsibility can also be minimized. In fact it is the case that if $\mathit{PassiveRespAntBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s)$ and C' is a superset of C, then $\mathit{PassiveRespAntBy}(C', \vec{f}'_C, \vec{f}_{\overline{C'}}, \varphi, s)$ (where the individual strategies are the same, i.e. $(\vec{f}_C, \vec{f}_{\overline{C}}) = (\vec{f}_{C'}, \vec{f}_{\overline{C'}})$). Thus with our definition of passive responsibility anticipation, it is indeed useful to focus on the minimal coalitions that have passive responsibility.

are as specified above by f_{ag}; and $\vec{f}_{\overline{C_1}}$ to denote the strategy of $\overline{C_1} = \{waiter, guard\}$, where, again, the individual strategies of *waiter* and *guard* are as specified above by f_{ag}.

In the actual scenario, both killers poison the drink in the first tick and the waiter serves it in the second. At all remaining ticks, the agents choose the other move, so the guard never intervenes. We can show that the coalition $C_1 = \{kllr1, kllr2\}$ is passively responsible for the attempted murder by strategies $\vec{f}_{C_1}$ and $\vec{f}_{\overline{C_1}}$. The waiter's move is a cause, but if she does not serve, *kllr2* can still deliver on the third tick, so the waiter cannot prevent the outcome. Thus she is not passively responsible by the strategies $\vec{f}_{C_2}$ and $\vec{f}_{\overline{C_2}}$, where $C_2 = \{waiter\}$. The guard, however, could have blocked delivery and thus prevented the outcome. We can show that by strategies $\vec{f}_{C_3}$ and $\vec{f}_{\overline{C_3}}$, the coalition $C_3 = \{guard\}$ is also passively responsible. The following figure shows passive responsibility attributions based on the mentioned strategies.

Coalition	**CausalResp**	**PassiveResp**	**Comb**
$\{kllr1, kllr2\}$	✓	✓	✓
$\{waiter\}$	✓	✗	✗
$\{guard\}$	✗	✓	✗

Figure 5.1: Responsibilities in the attempted murder scenario.

Causal Responsibility Attribution. While the above notions of strategic responsibility account for the choices made by the coalition and their consequences, they do

not capture the causal contributions to the outcome. For example, in the above scenario, both $\{kllr1, kllr2\}$ and $\{guard\}$ are considered passively responsible, yet this attribution overlooks their causal contribution to the attempted murder. To address this, we now define causal responsibility. Since actual causality is a retrospective notion, this is also defined relative to a history s, and is thus an ex post notion.

Definition 5.6.7 (Causal Responsibility Attribution)**.**

$$\begin{aligned}
&\mathit{CausRespAttrib}(C, \varphi, s) \stackrel{\text{def}}{=} \\
&\exists cc.\, \mathit{CompleteMinMovesCausalChain}(cc, \varphi, s) \wedge \\
&\forall ag.\, ag \in C \supset \exists a, ts.(a, ts) \in cc \wedge \mathit{AgentMove}(a, ag) \neq \mathit{wait}.
\end{aligned}$$

Thus, coalition C is causally responsible for effect φ in situation s iff all agents inside C contributed to the effect using a non-*wait* move in at least one (and the same) complete minimal-moves causal chain.

Using this, we define a combined notion of responsibility.

Definition 5.6.8 (Passive Combined Resp. Attribution)**.**

$$\begin{aligned}
&\mathit{PassiveCombRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s) \stackrel{\text{def}}{=} \\
&\mathit{CausRespAttrib}(C, \varphi, s) \wedge \mathit{ConsStrategySit}(C, s, \vec{f}_C, S_0) \wedge \\
&\mathit{ConsStrategySit}(\overline{C}, s, \vec{f}_{\overline{C}}, S_0) \wedge \\
&\mathit{PassiveRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s).
\end{aligned}$$

Thus, C has combined responsibility for φ by $\vec{f}_C$ and $\vec{f}_{\overline{C}}$ in s iff it is causally responsible for φ in s, s is consistent with both $\vec{f}_C$ and $\vec{f}_{\overline{C}}$ starting in the initial situation S_0, and C is passively responsible for φ by $\vec{f}_C$ and $\vec{f}_{\overline{C}}$ in s. Note that, there are cases where C has passive strategic responsibility for φ, but it is not causally responsible for it.

Returning to our example, we can show that $\{kllr1, kllr2\}$ and $\{waiter\}$ are causally responsible, while $\{guard\}$ is not, allowing us to distinguish between coalitions that merely enable outcomes and those that bring them about. $\{waiter\}$'s causal involvement however lacks strategic intent given the strategies of the others agents, which is consistent with her innocent bystander role. See figure 5.1.

5.7 Properties of Responsibility

• **Temporal Consistency:** A direct consequence of Def. 5.6.5 and 5.6.6 is that if a coalition is causally responsible in anticipation, then—if the anticipated structure unfolds—it is also responsible in attribution.

Corollary 5.7.1 (From Anticipation to Attribution)**.**

$$\mathcal{D} \models \forall C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s.\ (\mathit{PassiveRespAntBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s)$$
$$\wedge\ \exists s^*.\ s^* > s \wedge \varphi[s^*] \wedge \mathit{ConsStrategySit}(C, s^*, \vec{f}_C, s)$$
$$\wedge\ \mathit{ConsStrategySit}(\overline{C}, s^*, \vec{f}_{\overline{C}}, s))$$
$$\supset\ \mathit{PassiveRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s).$$

This also (trivially) holds in the other direction.

Corollary 5.7.2 (From Attribution to Anticipation)**.**

$$\mathcal{D} \models \forall C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s.\ \mathit{PassiveRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s)$$
$$\supset\ \exists s'.\ s' < s \wedge \mathit{PassiveRespAntBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s').$$

• **Non-Redundancy of Causal Responsibility:**

Theorem 5.7.3 (Causal vs. Strategic Responsibility)**.**

$$\mathcal{D} \not\models \forall C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s.\ \mathit{PassiveRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s)$$
$$\supset\ \mathit{CausRespAttrib}(C, \varphi, s),$$
$$\mathcal{D} \not\models \forall C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s.\ \mathit{CausRespAttrib}(C, \varphi, s)$$
$$\supset\ \mathit{PassiveRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s).$$

Proof Sketch: By counterexample; see attempted murder example above. □

• **Responsibility Persistence:** If a coalition C has passive responsibility for φ by

$\vec{f}_C$ and $\vec{f}_{\overline{C}}$ in s, then it will retain this responsibility in all subsequent situations s' if both C and $\overline{C}$ keep following these strategies from s to s', and if φ holds in s'.

Theorem 5.7.4 (Persistence of Passive Responsibility)**.**

$$\begin{aligned}\mathcal{D} \models \forall C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s, s'.[&\mathit{PassiveRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s)\\ &\wedge s < s' \wedge \mathit{ConsStrategySit}(C, s', \vec{f}_C, s)\\ &\wedge \mathit{ConsStrategySit}(C, s', \vec{f}_{\overline{C}}, s) \wedge \varphi[s']]\\ &\supset \mathit{PassiveRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s').\end{aligned}$$

Proof Sketch: Follows from the antecedent and Definitions 5.6.5 and 5.6.6. □

Moreover, if in addition φ remains true in all situations in between s and s', then the above persistence result can be extended for combined responsibility as well.

Corollary 5.7.5 (Persistence of Combined Responsibility)**.**

$$\begin{aligned}\mathcal{D} \models \forall C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s, s'.&\\ [&\mathit{PassiveCombRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s)\\ &\wedge s < s' \wedge \mathit{ConsStrategySit}(C, s', \vec{f}_C, s)\\ &\wedge \mathit{ConsStrategySit}(C, s', \vec{f}_{\overline{C}}, s)\\ &\wedge \forall a^*, s^*.\ (s \leq do(a^*, s^*) \leq s' \supset \varphi[s^*])]\\ &\supset \mathit{PassiveCombRespAttribBy}(C, \vec{f}_C, \vec{f}_{\overline{C}}, \varphi, s').\end{aligned}$$

Proof Sketch: Follows from Definitions 5.4.6, 5.6.7, and 5.6.8, and Theorems 5.5.3

and 5.7.4. □

5.8 Conclusion

We proposed an account of causation as well as causal, strategic, and combined responsibility attribution in a synchronous game-theoretic multiagent logic framework. Our proposal builds on Batusov and Soutchanski's [21] original formulation of achievement causality in the situation calculus. The relationship between that framework and Halpern and Pearl's intervention-based counterfactual causality [20] was also formally examined in that work. Closely related to our work is the preliminary study in [30], which formalizes actual cause in the SCSGS; here we refine on this by defining complete causal chains and using these to formalize responsibility. To our knowledge, the only other work linking responsibility to actual causation is [1], where degrees of responsibility are defined in terms of the number of changes required to avoid the outcome.

Our proposal is nevertheless limited in many ways. We only dealt with achievement causation and considered objective responsibility exclusively. While we handled the responsibility of coalitions, we did not consider how responsibility/blame should be ultimately distributed between the members of the coalition. There are many philosophical puzzles, such as the bystander effect and the circle-of-blame, that need to be settled before such attribution can be formalized. In the future, it would be

interesting to study maintenance causation in SCSGS. Also, responsibility attribution should account for the knowledge of the agent, which requires the integration of epistemic logic with the current proposal. To rule out accidental effects, one must integrate conative logic and notions of goals and intentions with responsibility. This would allow one to distinguish responsibility incurred due to intentional actions and accidental ones. Further, considering obligations and deontic logic might shed some light on the bystander effect, e.g., by stipulating that due to her obligations, the day-care worker should be held more strongly responsible than all other bystanders when it comes to the muddy child. Finally, in the future, it would be interesting to look into the practical aspects of this research.

5.9 Acknowledgements

We would like to thank the anonymous reviewers for helping us improve this paper. This work is partially supported by the National Science and Engineering Research Council of Canada, by the University of Regina, and by York University.

Chapter 5 References

[1] Hana Chockler and Joseph Y. Halpern. "Responsibility and Blame: A Structural-Model Approach". In: *Journal of Artificial Intelligence Research* 22 (2004), pp. 93–115.

[2] Emiliano Lorini and François Schwarzentruber. "A logic for reasoning about counterfactual emotions". In: *Artif. Intell.* 175.3-4 (2011), pp. 814–847.

[3] Emiliano Lorini, Dominique Longin, and Eunate Mayor. "A logical analysis of responsibility attribution: emotions, individuals and collectives". In: *J. Log. Comput.* 24.6 (2014), pp. 1313–1339.

[4] Alexandru Baltag, Ilaria Canavotto, and Sonja Smets. "Causal Agency and Responsibility: A Refinement of STIT Logic". In: *Logic in High Definition: Trends in Logical Semantics*. Ed. by Alessandro Giordani and Jacek Malinowski. 2021, pp. 149–176.

[5] Aldo Iván Ramírez Abarca and Jan M. Broersen. "A Stit Logic of Responsibility". In: *21st International Conference on Autonomous Agents and Multiagent*

Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022. Ed. by Piotr Faliszewski, Viviana Mascardi, Catherine Pelachaud, and Matthew E. Taylor. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2022, pp. 1717–1719.

[6] Nils Bulling and Mehdi Dastani. "Coalitional Responsibility in Strategic Settings". In: *Proceedings of the 14th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA)*. Vol. 8143. Lecture Notes in Computer Science. Springer, 2013, pp. 172–189.

[7] Vahid Yazdanpanah, Mehdi Dastani, Wojciech Jamroga, Natasha Alechina, and Brian Logan. "Strategic Responsibility Under Imperfect Information". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*. Ed. by Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 592–600.

[8] Timothy Parker, Umberto Grandi, and Emiliano Lorini. "Anticipating Responsibility in Multiagent Planning". In: *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*. Ed. by Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein,

and Roxana Radulescu. Vol. 372. Frontiers in Artificial Intelligence and Applications. IOS Press, 2023, pp. 1859–1866.

[9] Giuseppe De Giacomo, Emiliano Lorini, Timothy Parker, and Gianmarco Parretti. "Responsibility Anticipation and Attribution in LTLf". In: *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, 16-22 August 2025*. ijcai.org, 2025.

[10] Matthew Braham and Martin van Hees. "An Anatomy of Moral Responsibility". In: *Mind* 121.483 (2012), pp. 601–634.

[11] Emiliano Lorini and Roland Mühlenbernd. "The Long-Term Benefits of Following Fairness Norms under Dynamics of Learning and Evolution". In: *Fundam. Informaticae* 158.1-3 (2018), pp. 121–148.

[12] Christel Baier, Florian Funke, and Rupak Majumdar. "A Game-Theoretic Account of Responsibility Allocation". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. Ed. by Zhi-Hua Zhou. ijcai.org, 2021, pp. 1773–1779.

[13] Pavel Naumov and Jia Tao. "Two Forms of Responsibility in Strategic Games". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. Ed. by Zhi-Hua Zhou. ijcai.org, 2021, pp. 1989–1995.

[14] Pavel Naumov and Jia Tao. "Counterfactual and seeing-to-it responsibilities in strategic games". In: *Ann. Pure Appl. Log.* 174.10 (2023), p. 103353.

[15] Qi Shi. "Responsibility in Extensive Form Games". In: *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, February 20-27, 2024, Vancouver, Canada*. Ed. by Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan. AAAI Press, 2024, pp. 19920–19928.

[16] Nicole A. Vincent. "A Structured Taxonomy of Responsibility Concepts". In: *Moral Responsibility: Beyond Free Will and Determinism*. Ed. by Nicole A. Vincent, Ibo van de Poel, and Jeroen van den Hoven. Dordrecht: Springer Netherlands, 2011, pp. 15–35.

[17] Harry G. Frankfurt. "Alternate possibilities and moral responsibility". In: *The Journal of Philosophy* 66.23 (1969), pp. 829–839.

[18] Joseph Y. Halpern. "Axiomatizing Causal Reasoning". In: *Journal of Artificial Intelligence Research* 12 (2000), pp. 317–337.

[19] Giuseppe De Giacomo, Yves Lespérance, and Adrian R. Pearce. "Situation Calculus Game Structures and GDL". In: *ECAI*. 2016, pp. 408–416.

[20] Joseph Y. Halpern. *Actual Causality*. MIT Press, 2016. ISBN: 978-0-262-03502-6.

[21] Vitaliy Batusov and Mikhail Soutchanski. "Situation Calculus Semantics for Actual Causality". In: *Proceedings of the Thirty-Second AAAI Conference on*

Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018, pp. 1744–1752.

[22] Shakil M. Khan and Yves Lespérance. "Knowing Why - On the Dynamics of Knowledge about Actual Causes in the Situation Calculus". In: *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*. Ed. by Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé. ACM, 2021, pp. 701–709.

[23] John McCarthy and Patrick J. Hayes. "Some Philosophical Problems from the Standpoint of Artificial Intelligence". In: *Machine Intelligence* 4 (1969), pp. 463–502.

[24] Raymond Reiter. *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. Cambridge, MA, USA: MIT Press, 2001. ISBN: 9780262182188.

[25] Hector J. Levesque, Fiora Pirri, and Raymond Reiter. "Foundations for the Situation Calculus". In: *Electronic Transactions on Artificial Intelligence (ETAI)* 2 (1998), pp. 159–178.

[26] Javier Pinto. "Concurrent Actions and Interacting Effects". In: *KR*. 1998, pp. 292–303.

[27] Shakil M. Khan and Yves Lespérance. "Infinite Paths in the Situation Calculus: Axiomatization and Properties". In: *Principles of Knowledge Representation*

and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016. Ed. by Chitta Baral, James P. Delgrande, and Frank Wolter. AAAI Press, 2016, pp. 565–568.

[28] Giuseppe De Giacomo, Yves Lespérance, and Matteo Mancanelli. "Situation Calculus Temporally Lifted Abstractions for Generalized Planning". In: *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*. Ed. by Toby Walsh, Julie Shah, and Zico Kolter. AAAI Press, 2025, pp. 14848–14857.

[29] Ned Hall. "Two Concepts of Causation". In: *Causation and Counterfactuals*. Ed. by John Collins, Ned Hall, and L. A. Paul. MIT Press, 2004, pp. 225–276.

[30] MohammadHossein Karimian, Shakil M. Khan, and Yves Lespérance. "On the Semantics of Actual Causality in Situation Calculus Concurrent Game Structures". In: *38th Canadian Conference on Artificial Intelligence, Canadian AI 2025, Calgary, AB, Canada, May 26-29, 2025, Proceedings*. Canadian Artificial Intelligence Association, 2025.

[31] Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque. "ConGolog, A Concurrent Programming Language based on the Situation Calculus". In: *Artificial Intelligence* 121.1-2 (2000), pp. 109–169.

Chapter 6

Discussion and Conclusion

6.1 Summary of Contributions

In this thesis, I studied actual causation and causal and strategic notions of responsibility in *situation calculus concurrent game structures (SCSGS)*, a first-order extension of concurrent game structures with an action-theoretic underpinning. In the following, I list the core novel contributions of this thesis.

- **A formalization of actual causality in SCSGS.** I gave a full account of actual causation for synchronous concurrent multi-agent domains, extending previous attempts that focused on turn-taking multi-agent systems exclusively [12, 28]. The account captures primary (direct) and indirect causes and builds causal chains inductively. For this, I started by utilizing previous proposals on actual cause in the situation calculus to identify the tick actions that are causes, as well as the associated causal chains. I then refined this causal chain to untangle and identify all the minimal-moves causal chains. By retaining

only necessary moves and replacing inessential ones with *wait*, I thus yield only the minimal sets of moves, each of which might have caused the effect. This minimization lets me avoid over-determination.

- **Properties of actual cause in SCSGS.** I prove that the framework does not misclassify preempted actions as causes. My minimality condition also avoids over-determination. I also study the persistence of actual causes in this framework. To some extent, these results establish the intuitive correctness of my proposal.

- **A formalization of strategic and causal responsibility in SCSGS.** I redefined previously proposed notions of active responsibility, passive responsibility anticipation, and passive responsibility attribution, now extended to deal with coalitions of agents. For this, I used the SCSGS enriched with paths in the situation calculus and a previously proposed notion of CanForceBy (i.e., an agent being able to force a Linear Temporal Logic property by following a strategy), which I also extended to deal with coalitions (multi-agent CanForceBy).

  Moreover, based on my formalization of actual cause in SCSGSs, I proposed a novel notion of causal responsibility attribution. This allows us to formally capture the coalitions that were responsible for causally contributing to the outcome. I also defined a notion of combined (i.e., both passive strategic and causal) responsibility.

- **Properties of causal and strategic responsibility in SCSGS.** I proved

some interesting general properties, showing how causal and strategic responsibility are related and investigating the conditions under which various types of responsibility persist. I also studied temporal consistency between ex post and ex ante variants of passive responsibility.

- **Formal Examples.** To demonstrate the proposed theory of actual cause in SC-SGS, I formalized some variants of the famous bottle example and showed how it avoids misclassifying irrelevant actions/moves as causes and does not suffer from common causal problems such as preemption and over-determination.

To illustrate the usefulness of this new proposal of causal responsibility, I formalized the attempted murder example, where the value of causal responsibility becomes quite evident. It allowed me to show that causal, passive, and combined notions of responsibility have different extensions: a coalition can be causally responsible without being passively responsible and vice versa.

6.2 Conclusion and Future Work

Halpern and Pearl's causal models are not based on proper action theories and do not assume any ordering of event occurrence. Recent proposal on action-theoretic formalisms of actual cause, on the other hand, assumes linear scenarios. In contrast, my proposal allows one to model more realistic scenarios where agents can act synchronously. I also studied causal responsibility within this setting.

The proposals in this thesis are nevertheless limited in many ways. I only dealt

with achievement causation and considered objective responsibility. While I handled the responsibility of coalitions, I did not consider how responsibility/blame should be ultimately distributed between the members of the coalition. There are many philosophical puzzles, such as the bystander effect and the circle-of-blame, that need to be settled before such attribution can be formalized.

In the future, it would be interesting to study maintenance causation in SCSGSs. Also, responsibility attribution should account for the knowledge (or lack thereof) of the agent, which requires the integration of epistemic logic with the current proposal. To rule out accidental effects, one must also integrate conative logic and notions of goals and intentions with responsibility. This would allow one to distinguish responsibility incurred due to intentional actions and accidental ones. Further, considering obligations and deontic logic might shed some light on the bystander effect, e.g., by stipulating that the daycare worker should be held strongly responsible than all other bystanders when it comes to the muddy child. Finally, it would be interesting to look into the practical aspects of this research.

General References

[1] Judea Pearl. *On the Definition of Actual Cause*. Tech. rep. R-259. University of California Los Angeles, 1998.

[2] Joseph Y. Halpern and Judea Pearl. "Causes and Explanations: A Structural-Model Approach. Part I: Causes". In: *The British Journal for the Philosophy of Science* 56.4 (2005), pp. 843–887.

[3] Joseph Y. Halpern. *Actual Causality*. MIT Press, 2016. ISBN: 978-0-262-03502-6.

[4] Ned Hall. "Two Concepts of Causation". In: *Causation and Counterfactuals*. Ed. by John Collins, Ned Hall, and L. A. Paul. MIT Press, 2004, pp. 225–276.

[5] Ned Hall. "Structural Equations and Causation". In: *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition* 132.1 (2007), pp. 109–136.

[6] Vitaliy Batusov and Mikhail Soutchanski. "Situation Calculus Semantics for Actual Causality". In: *Proceedings of the Thirty-Second AAAI Conference on*

Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, 2018, pp. 1744–1752.

[7] Vahid Yazdanpanah, Mehdi Dastani, Wojciech Jamroga, Natasha Alechina, and Brian Logan. "Strategic Responsibility Under Imperfect Information". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*. Ed. by Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 592–600.

[8] Giuseppe De Giacomo, Emiliano Lorini, Timothy Parker, and Gianmarco Parretti. "Responsibility Anticipation and Attribution in LTLf". In: *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, 16-22 August 2025*. ijcai.org, 2025.

[9] David Lewis. "Causation". In: *Journal of Philosophy* 70.17 (1973), pp. 556–567.

[10] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[11] Mark Hopkins and Judea Pearl. "Causality and Counterfactuals in the Situation Calculus". In: *Journal of Logic and Computation* 17.5 (2007), pp. 939–953.

[12] Hana Chockler and Joseph Y. Halpern. "Responsibility and Blame: A Structural-Model Approach". In: *Journal of Artificial Intelligence Research* 22 (2004), pp. 93–115.

[13] Roberto Ciuni and Rosja Mastop. "Attributing Distributed Responsibility in STIT Logic". In: *Logic, Rationality, and Interaction — 2nd International Workshop (LORI 2009), Proceedings*. Ed. by Xiangdong He, J. F. Horty, and Eric Pacuit. Vol. 5834. Berlin / Heidelberg: Springer, 2009, pp. 66–75. DOI: 10.1007/978-3-642-04893-7_6.

[14] Emiliano Lorini, Dominique Longin, and Eunate Mayor. "A logical analysis of responsibility attribution: emotions, individuals and collectives". In: *J. Log. Comput.* 24.6 (2014), pp. 1313–1339.

[15] David Hume. *A Treatise of Human Nature*. Ed. by L. A. Selby-Bigge and P. H. Nidditch. 2nd (revised by Nidditch). Book I, Part III (*Of knowledge and probability*). Repr. 1978. Oxford: Clarendon Press, 1739.

[16] David Hume. *An Enquiry concerning Human Understanding*. Ed. by L. A. Selby-Bigge and P. H. Nidditch. 3rd (revised by Nidditch). Especially Section VII: *Of the idea of necessary connexion*. Repr. 1975. Oxford: Clarendon Press, 1748.

[17] MohammadHossein Karimian, Shakil M. Khan, and Yves Lespérance. "On the Semantics of Actual Causality in Situation Calculus Concurrent Game Structures". In: *38th Canadian Conference on Artificial Intelligence, Canadian AI*

2025, Calgary, AB, Canada, May 26-29, 2025, Proceedings. Canadian Artificial Intelligence Association, 2025.

[18] Joseph Y. Halpern. "A Modification of the Halpern-Pearl Definition of Causality". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. Ed. by Qiang Yang and Michael J. Wooldridge. AAAI Press, 2015, pp. 3022–3033.

[19] Gregor Gößler, Oleg Sokolsky, and Jean-Bernard Stefani. "Counterfactual Causality from First Principles?" In: *Proceedings 2nd Intl. Workshop on Causal Reasoning for Embedded and safety-critical Systems Technologies, CREST@ETAPS 2017*. 2017, pp. 47–53.

[20] Shakil M. Khan and Mikhail Soutchanski. "Necessary and Sufficient Conditions for Actual Root Causes". In: *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*. Ed. by Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang. Vol. 325. Frontiers in Artificial Intelligence and Applications. IOS Press, 2020, pp. 800–808.

[21] Maksim Gladyshev, Natasha Alechina, Mehdi Dastani, Dragan Doder, and Brian Logan. "Temporal Causal Reasoning with (Non-Recursive) Structural

Equation Models". In: *AAAI Proceedings / AAAI Press* 39.14 (2025). Also available as arXiv:2501.10190.

[22] Joseph Y. Halpern and Spencer Peters. "Reasoning About Causal Models with Infinitely Many Variables". In: *Proceedings of the 36th AAAI Conference on Artificial Intelligence*. 2022.

[23] Carlos Aguilera-Ventura, Xinghan Liu, Emiliano Lorini, and Dmitry Rozplokhas. "A Non-Interventionist Approach to Causal Reasoning Based on Lewisian Counterfactuals". In: *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2025)*. Montreal, Canada: IJCAI Organization, 2025, 479–?

[24] Sander Beckers and Joost Vennekens. "A Principled Approach to Defining Actual Causation". In: *Synthese* 195.2 (2018), pp. 835–862.

[25] Alexander Bochman. "A Logic For Causal Reasoning". In: *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*. Ed. by Georg Gottlob and Toby Walsh. Morgan Kaufmann, 2003, pp. 141–146.

[26] Alexander Bochman. "Actual Causality in a Logical Setting". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Ed. by Jérôme Lang. ijcai.org, 2018, pp. 1730–1736.

[27] John Leslie Mackie. "Causes and Conditions". In: *American Philosophical Quarterly* 2.4 (1965), pp. 245–264.

[28] Shakil M. Khan and Yves Lespérance. "Knowing Why - On the Dynamics of Knowledge about Actual Causes in the Situation Calculus". In: *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*. Ed. by Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé. ACM, 2021, pp. 701–709.

[29] Shakil M. Khan and Maryam Rostamigiv. "On Explaining Agent Behaviour via Root Cause Analysis: A Formal Account Grounded in Theory of Mind". In: *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland*. Ed. by Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu. Vol. 372. Frontiers in Artificial Intelligence and Applications. IOS Press, 2023, pp. 1239–1247.

[30] Shakil M. Khan, Yves Lespérance, and Maryam Rostamigiv. "Reasoning about Actual Causes in Nondeterministic Domains". In: *Proceedings of the 39th Annual AAAI Conference on Artificial Intelligence (AAAI-25), February 25 - March 4, 2025, Philadelphia, Pennsylvania, USA*. AAAI Press, 2025.

[31] Shakil M. Khan, Yves Lespérance, and Maryam Rostamigiv. "Reasoning About Causal Knowledge in Nondeterministic Domains". In: *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2025)*.

Montreal, Canada: IJCAI Organization, 2025, pp. 4553–4561. DOI: 10.24963/ijcai.2025/507.

[32] Emiliano Lorini and François Schwarzentruber. "A logic for reasoning about counterfactual emotions". In: *Artif. Intell.* 175.3-4 (2011), pp. 814–847.

[33] Alexandru Baltag, Ilaria Canavotto, and Sonja Smets. "Causal Agency and Responsibility: A Refinement of STIT Logic". In: *Logic in High Definition: Trends in Logical Semantics*. Ed. by Alessandro Giordani and Jacek Malinowski. 2021, pp. 149–176.

[34] Aldo Iván Ramírez Abarca and Jan M. Broersen. "A Stit Logic of Responsibility". In: *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*. Ed. by Piotr Faliszewski, Viviana Mascardi, Catherine Pelachaud, and Matthew E. Taylor. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2022, pp. 1717–1719.

[35] Nuel Belnap, Michel Perloff, and Ming Xu. *Facing the Future: Agents and Choices in our Indeterministic World*. Oxford University Press, 2001. ISBN: 9780195138788.

[36] Emiliano Lorini and Roland Mühlenbernd. "The Long-Term Benefits of Following Fairness Norms under Dynamics of Learning and Evolution". In: *Fundam. Informaticae* 158.1-3 (2018), pp. 121–148.

[37] Harry G. Frankfurt. “Alternate Possibilities and Moral Responsibility”. In: *Journal of Philosophy* 66.23 (1969), pp. 829–839. DOI: 10.2307/2023833.

[38] Timothy Parker, Umberto Grandi, and Emiliano Lorini. “Anticipating Responsibility in Multiagent Planning”. In: *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*. Ed. by Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu. Vol. 372. Frontiers in Artificial Intelligence and Applications. IOS Press, 2023, pp. 1859–1866.

[39] Nils Bulling and Mehdi Dastani. “Coalitional Responsibility in Strategic Settings”. In: *Proceedings of the 14th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA)*. Vol. 8143. Lecture Notes in Computer Science. Springer, 2013, pp. 172–189.

[40] Christel Baier, Florian Funke, and Rupak Majumdar. “A Game-Theoretic Account of Responsibility Allocation”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. Ed. by Zhi-Hua Zhou. ijcai.org, 2021, pp. 1773–1779.

[41] John McCarthy and Patrick J. Hayes. “Some Philosophical Problems from the Standpoint of Artificial Intelligence”. In: *Machine Intelligence* 4 (1969), pp. 463–502.

[42] Giuseppe De Giacomo, Yves Lespérance, and Adrian R. Pearce. "Situation Calculus Game Structures and GDL". In: *ECAI*. 2016, pp. 408–416.

[43] Raymond Reiter. *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. Cambridge, MA, USA: MIT Press, 2001. ISBN: 9780262182188.

[44] Shakil M. Khan and Yves Lespérance. "Epistemic Causes and Effects in the Situation Calculus". In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2021)*. IFAAMAS, 2021, pp. 683–691.