TOWARDS ROOT CAUSE ANALYSIS IN HYBRID DYNAMIC DOMAINS

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

Asim Mehmood

Regina, Saskatchewan

December 2024

Abstract

Reasoning about actual causes of observed effects is fundamental to the study of rationality. As such, this important problem has been studied since the time of Aristotle, with formal mathematical accounts emerging recently. We live in a world where change due to actions can be both discrete and continuous, i.e., hybrid. Yet, while there has been extensive research on actual primary and indirect causes in discrete dynamic domains, only few recent studies address causation in such hybrid domains. Building on recent progress, in this thesis I propose a first definition of primary cause in a hybrid temporal action-theoretic framework. My proposal is limited to primitive observations/effects. I also show how a variant of my definition can be interpreted from a counterfactual perspective and hint how the account can be modified to work with conjunctive/disjunctive effects. My proposal is set within a hybrid variant of the situation calculus. I show that my formalization has some basic intuitive properties.

Acknowledgments

First and foremost, I would like to thank God for giving me the strength and perseverance to complete this work. I extend my deepest gratitude to my supervisor, Dr. Shakil M. Khan, for his unwavering support, insightful guidance, and continuous encouragement. His mentorship has been invaluable. My heartfelt thanks go to my family for their love, support, and belief in me. I also wish to acknowledge Dr. Allen Herman for his role as the external examiner during my defense and Dr. Sandra Zilles for serving as a committee member. Furthermore, I extend my sincere thanks to Dr. Lisa Fan for her valuable comments on my thesis. Finally, I am thankful to the Faculty of Graduate Studies and Research at the University of Regina for funding my research, as well as to the Natural Sciences and Engineering Research Council of Canada for their support through grants awarded to my exceptional supervisor.

Contents

List of Figures

Index to Symbols and their Definitions

Transparency Statement

I used Grammarly to correct grammatical errors and ChatGPT to validate and paraphrase text, as well as to further explore certain topics in the literature review.

My supervisor(s) and supervisory committee have approved the use of the above technologies for the described purposes. I confirm that no AI-technologies other than those listed above have been used to prepare this thesis. I acknowledge that AI-technologies may produce output that is biased, discriminatory, incomplete, or inaccurate and that I have taken the necessary steps to address this. I acknowledge that I am solely responsible for maintaining the accuracy and academic integrity of this thesis.

Chapter 1

Introduction

Causality, the relationship between cause and effect, is a fundamental concept critical to our understanding of the world. Philosophers categorize causality into two types: type-level or general causality and token-level or actual causality. Type-level causality refers to general causal mechanisms describing the relationship between events (e.g., physical inactivity leads to health problems), whereas actual causality addresses the causes of a specific observed event given a history of the evolution of the world (e.g., why the train did not arrive at 8 am given certain actions of the train engineers). It is not difficult to see the importance of actual causation (or just "causation" henceforth) in the study of rationality. Such information might be used by an artificial agent (i.e., an autonomous computational system that is reactive, proactive, and intentional) to decide on what to do next, e.g., by recognizing the intentions of another agent it is interacting with, or repair its plans by analyzing what went wrong in the previous one. Causation is also important for explainable

artificial intelligence, which requires users of artificial intelligence to on demand ask for explaining the choices/decisions made by an agent. Unlike predictive models, which forecast future events based on patterns or past histories, causal explanations aim to understand why and how events occur in a cause-and-effect relationship.

Establishing actual causality is complex due to multiple influencing factors or variables, making it challenging to isolate the exact cause. Philosophers and scientists have been exploring causal laws since the time of Aristotle, and still, there is no universally agreed-upon definition that can cover all scenarios [20].

Based on Pearl's original work [45, 46], Halpern and Pearl, among others [18, 22, 11, 24, 25, 19, 20], have extensively studied actual causation and advanced this field significantly. Halpern and Pearl's approach is based on structural equations models (SEM) [55] and follows the Humean counterfactual definition of causation [27]. This definition states that "an outcome B is caused by an event A" is the same as saying that "had A never occurred, B would never have existed." However, this approach suffers from the problem of preemption, where another event A' that occurred after A in the original history could still cause B in the absence of A (this effect of A' is said to be preempted by that of A in the original history). Halpern and Pearl avoid preemption by performing selective counterfactual analysis and suspending some of the model's mechanisms. Despite its practical applications, their approach has been criticized to be problematic [32], has limited expressiveness, and lacks clear guidelines for model selection in causal analysis [24, 25, 16]. To deal with these, researchers have attempted to extend it with additional features [34]. However, many limitations

still remain. For example, it is not clear how one can formalize various aspects of action-theoretic/dynamic frameworks there, such as non-persistent change supported by fluents, possible dependency between events, temporal order of event occurrence, etc.

To address this, recently, some researchers have focused on studying causation within more expressive action-theoretic frameworks, particularly the situation calculus [3, 4, 32]. The situation calculus allows one to formalize causation from the first-person's perspective through the study of epistemic causation (i.e., causal knowledge [30]), has proven to be useful for explaining agent behavior using causal analysis [31], and has the potential for defining important concepts like responsibility and blame [58].

1.1 Motivation and Specific Problem

A distinguishing feature of the real world is that change can be both discrete and continuous. For instance, when a loss-of-coolant accident occurs due to a pipe rupture, the associated nuclear power plant may not overheat immediately. Instead, such safety failures might happen gradually over time. Despite this "hybrid" nature of change in the real world, almost all of the work on actual causation has focused on defining causes within discrete domains. In fact, to the best of my knowledge, only one recent study addressed causation in hybrid systems causal models [23]. In that work, Halpern and Peters introduced an extension of structural-equation models, known

as generalized structural-equation models. These models can capture hybrid systems by permitting only specified interventions, potentially resulting in an infinite number of outcomes. To manage the complexity of analysis, the language for discussing these models is restricted to only explicitly reference countably many values and interventions. Despite improving on the expressivity of structural-equation-based causal models, it suffers from the inherent limitations of structural-equations based causal models as mentioned above.

Returning to our example, however, it would be obviously useful to understand the causes of such nuclear power plant core overheating given the logged events. Among other things, such information can be used to avoid future disasters. Thus causal analysis in hybrid domains is essential for gaining insights into system behavior to understand how discrete events and continuous processes interact to produce observed behavior. It can help identify the chain of events leading to a system failure, aiding in diagnosis and troubleshooting. One can predict how the system will respond to different inputs, which is essential for designing robust control strategies and ensuring system reliability. In some domains, such as autonomous vehicles or medical devices, understanding causation is not only important for technical reasons but also for legal and ethical considerations. For example, knowing the causes of accidents or failures can help assign responsibility and ensure accountability.

1.2 Contributions

Inspired by previous work on the action-theoretic formalization of actual causation [4, 30], in this thesis, I propose a formal account of actual causality in hybrid dynamic domains. My proposal is set within a recently developed hybrid variant of the situation calculus, namely the hybrid temporal situation calculus [1, 2]. I focus on actual primary causes and study causation relative to primitive fluents exclusively. My contributions are outlined below:

1. In discrete domains, I define counterfactual worlds and demonstrate that removing the primary cause may not always remove the effect (e.g., due to preempted actions).

2. I define a proper "causal setting" for hybrid domains, which includes a tuple consisting of a domain theory specifying the actions in the domain, a history of actions that occurred before the effect was observed (called the scenario), and the effect for which we aim to identify the causes in the given scenario. Then I introduce a definition of the primary achievement cause relative to a causal setting with effects involving a primitive temporal fluent. This involves addressing the challenge of identifying the achievement situation of an effect to determine causes while considering the temporal order of events/actions.

3. I present preliminary definitions of primary cause for compound cases, where the effect involves a conjunction or a disjunction of primitive temporal fluents.

4. I prove some basic properties of my formalization of primary achievement cause.

5. I propose a new definition of primary cause in hybrid domains that was developed to study causes from a counterfactual perspective. I show that this definition is equivalent to the one in 2 above.

6. I extend my counterfactual analysis above from discrete to hybrid domains and formalize a notion of defused situations, which removes the cause along with preempted causes. Using this, I show that without the cause, the effect does not follow in such a defused situation, unless the required conditions were already initially true. This allows me to show how my proposal can be linked to a counterfactual interpretation of causes.

7. To illustrate these definitions, I give some formal examples and prove various properties of these domains.

A preliminary version of some of my proposals in contributions 2 to 4 above appeared in the 37th Canadian Conference on Artificial Intelligence (Canadian AI 2024), where our paper won the Best Student Paper award; see [42].

1.3 Thesis Organization

The thesis is organized as follows. In the next chapter, I survey the literature, providing a comprehensive overview of the existing research and theories related to causality. In Chapter 3, I introduce my base framework, the situation calculus [52],

and the recently proposed hybrid temporal situation calculus [1]. I recap previous work on actual causation within the situation calculus for discrete domains and highlight relevant theories that have influenced my approach. I also propose a definition of counterfactual situations in the situation calculus. This part is novel to this thesis. In Chapter 4, I propose a definition of primary achievement cause for effects involving primitive temporal fluents, prove some properties of my definition, and illustrate the intuition of my work with examples. In Chapter 5, I give another definition of the primary cause under the same assumptions, show how this new definition can characterize my proposal in counterfactual terms, and give formal examples for better understanding. Finally, in Chapter 6, I conclude the thesis with a discussion of results and possible future work.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, I review previous work on modeling actual causality. I begin by introducing two popular approaches to causation, the regularity and counterfactual theories. I then examine a pioneering approach based on Structural-Equations, discussing its implications and limitations. Following this, I review research conducted within a more expressive action-theoretic framework—the situation calculus—which effectively models dynamic domains, noting how it enriches causal analysis while being limited to discrete domains. Finally, I explain the concept of hybrid systems and discuss the only other work on causation in hybrid domains, highlighting its limitations.

2.2 Regularity Theories of Causation

According to David Hume's 18th-century regularity theory [27], causation is not grounded in a necessary connection between cause and effect but is instead based on the observation that certain events (causes) are consistently followed by other events (effects). Hume identified three key aspects of causation: constant conjunction (events are regularly associated), temporal succession (the cause precedes the effect), and the notion of necessity (which we infer from repeated observations rather than direct perception). For instance, when a hot coffee is spilled, burning of the hand often follows. This repeated succession leads us to form an association between cause and effect. We come to expect the effect when we encounter the cause, not due to any inherent causal power but because of the observed pattern of regularity. As Hume [27] famously remarked, "We only infer the existence of one thing from the appearance of the other and are led to conceive that there is some connection between them, but are never able to discover what that connection is."

Hume's regularity theory thus challenges the intuitive belief in a necessary connection between cause and effect and has profoundly influenced subsequent philosophical and scientific discussions. However, a potential concern could arise regarding scenarios, particularly hybrid temporal ones, where the effects may not be immediately observable. This issue is the central focus of my research.

Others have later proposed their formalization of causation based on Hume's work and came up with their own variant of regularity theory, e.g., [38].

2.3 Causes and INUS Conditions

Mackie [38] argued that causation is complex, involving multiple factors that may not be sufficient alone but are necessary within a set of conditions that together cause an effect. For example, a power plant needs fuel, a generator, cooling water, and a transmission system to generate electricity. None of these alone is enough, but together, they are sufficient to produce and distribute electricity. Also, the notion that a cause must be a necessary and sufficient preceding condition for an effect is problematic because multiple sufficient conditions often exist, and a cause might be part of one sufficient condition but replaceable by other factors in a different sufficient condition.

Building on Hume's regularity theory, Mackie [38] introduced the *INUS* condition to address this oversimplification in causal complexity, representing a significant refinement in the study of causation. The *INUS* condition stands for an "Insufficient but Necessary part of a condition which is itself Unnecessary but Sufficient" for the occurrence of a certain effect. In this analysis, for an event A to be considered a cause of event B, there must exist a combination of conditions X and Y such that $(A \wedge X) \vee Y$ is both necessary and sufficient for B. This implies that neither A nor X alone can bring about B, but A in conjunction with X can, and Y provides an alternative sufficient condition.

In a power plant scenario, fuel is an *INUS* condition for generating electricity. The fuel alone isn't enough; it also needs a generator, cooling water, and a transmission

system. While necessary in this scenario, electricity could also be generated using wind or solar power, making the specific fuel unnecessary in a broader sense. However, when combined with the other factors, the fuel is sufficient to generate electricity. This framework emphasizes the complexity and interconnectedness of causal relationships, challenging simpler views of causation that focus on direct, one-to-one effects, by highlighting how specific conditions contribute to an outcome in a non-redundant manner.

2.4 Counterfactual Theories of Causation

David Lewis [37] suggests that causation can be understood in terms of what would happen under different alternative conditions to what actually happened, i.e., counterfactual situations. Formally, an event A is said to cause an event B if and only if in the closest possible world where A does not occur, B also does not occur. The closest possible world is the one that is most similar to the actual world and where A does not occur.

Critics argue that counterfactual theories of causation are potentially unreliable due to the subjectivity in defining the "closest possible world", an overemphasis on counterfactual dependence while neglecting actual causal mechanisms, and challenges in applying the theory to complex systems with many interacting variables.

Defenders of the counterfactual theory argue that, despite criticisms of its potential subjectivity and complexity in real-world scenarios, the theory can be objectively

defined through rigorous criteria, integrated with other causal theories for a more comprehensive understanding, and remains practically valuable in applications such as legal reasoning and scientific modeling.

In fact, experimental results from psychology show that varying relevant counterfactual worlds while keeping the actual world events fixed strongly affect participants' causal judgments [44]. In contrast, keeping the counterfactual worlds constant and varying how the actual outcome was brought about much less influenced their causal judgments. This demonstrates that human causal judgments are inextricably linked to counterfactuals. This close interconnection has also been emphasized by researchers while studying causal responsibility [32] and causation in legal and moral reasoning [24]. Much of the work on causal models discussed next is based on counterfactual reasoning.

2.5 Causal Models based on Structural-Equations-Modeling

Halpern and Pearl [18, 19, 20, 21, 22] pioneered the formal modeling of actual causality based on the notion of counterfactuals, utilizing Structural-Equations Models to analyze causal relationships between variables. Structural-equations models use mathematical equations to define and analyze the causal relationships among multiple variables, including both direct and indirect effects. A structural-equations model consists of a pair $(\mathcal{S}, \mathcal{F})$, where $\mathcal{S}$ is the signature listing the variables and their possible values, and $\mathcal{F}$ is a set of equations. A signature $\mathcal{S}$ is a tuple $(\mathcal{U}, \mathcal{V}, \mathcal{R}, \mathcal{I})$

where $\mathcal{U}$ is the set of exogenous variables, $\mathcal{V}$ is the set of endogenous variables, $\mathcal{I}$ is a set of interventions, and $\mathcal{R}$ assigns each variable $\mathcal{Y} \in \mathcal{U} \cup \mathcal{V}$ a finite set of possible values $\mathcal{R}(\mathcal{Y})$. $\mathcal{F}$ specifies how the values of endogenous variables are determined by exogenous variables and other endogenous variables. Endogenous variables are the ones whose values are determined by other variables in the model, while exogenous variables are outside factors that affect the model but are not influenced by it. Interventions, involve setting the value of a variable, resulting in a modified structural equation model that reflects counterfactuals. For each endogenous variable X, there is an associated function $\mathcal{F}_X$ that determines its value based on other variables in $\mathcal{U} \cup \mathcal{V}$. Outcomes are determined by solving the structural equations given the context and interventions. For example, if the structural equation is given by $\mathcal{F}_X(u, y, z) = u - y$, then X is determined by the equation $X = U - Y$. If $U = 5$ and $Y = 3$, substituting these values into the equation gives $X = 5 - 3 = 2$.

According to Halpern and Pearl, an event X is considered an actual cause of another event Y if and only if:

1. There exists a causal chain from X to Y, meaning factor X influences Y.
2. There are no blocking paths preventing the flow of influence from X to Y, considering counterfactual reasoning to understand how changing one factor would affect the outcome.

They proposed the following conditions for an event to be the cause of another:

1. **Sufficiency**: X is a cause of Y if the occurrence of X is sufficient to bring

about Y.

2. **Necessity**: X is a cause of Y if the absence of X would have prevented Y.

3. **Proportionality**: X is a cause of Y if it increases the probability of Y.

They came up with three different definitions, each of which was later proven problematic (using counter-examples) by others. I will give their modified definition of the actual cause below:

Definition 2.5.1. *Let $\mathcal{U}$ and $\mathcal{V}$ be the sets of exogenous and endogenous variables, $(M, \bar{V}_{\mathcal{U}})$ be a causal setting, X be an endogenous variable, and V_X be the value of X (see [20] for details). The conjunction of primitive events $\bar{X} = \bar{V}_X$, short for $X_1 = V_{X_1} \wedge \cdots \wedge X_k = V_{X_k}$, is an actual cause in $(M, \bar{V}_U)$ of an HP query φ (whether one event is an actual cause of another event in a given causal model) if all the following conditions hold:*

1. *$(M, \bar{V}_U) \models (\bar{X} = \bar{V}_X)$ and $(M, \bar{V}_U) \models \varphi$.*

2. *There exists a set $\bar{W}$ (disjoint from $\bar{X}$) of variables in V with $(M, \bar{V}_U) \models (\bar{W} = \bar{V}_W)$ and a setting $\bar{V}'_X$ of variables $\bar{X}$ such that $(M, \bar{V}_U) \models [\bar{X} \leftarrow \bar{V}'_X, \bar{W} \leftarrow \bar{V}_W]\neg\varphi$.*

3. *No proper subconjunction of $(\bar{X} = \bar{V}_X)$ satisfies conditions 1 and 2.*

That is, a set of variables X is an actual cause of an outcome φ if both X and φ are true in the actual scenario. Changing X to different values while keeping other

related variables fixed must make φ false. Additionally, no proper subset of X can fulfill these conditions, indicating that X is essential for causing φ.

The limitations of the above definition can be illustrated through the well-known "bottle" example. In this scenario, Suzy and Billy each throw rocks at a bottle. Suzy's rock hits first, shattering the bottle, while Billy's rock would have shattered it if Suzy's had not. The situation is modeled using structural equations with five endogenous variables: ST (Suzy throws), SH (Suzy hits), BT (Billy throws), BH (Billy hits), and BS (bottle shatters). The relationships are defined as follows:

- ST = 1 (Suzy throws),
- BT = 1 (Billy throws),
- SH = ST (Suzy hitting depends on her throw),
- BH = BT $\wedge$ $\neg$SH (Billy hitting depends on his throw and Suzy missing),
- BS = SH $\vee$ BH (the bottle shatters if either Suzy or Billy hits it).

The modified definition identifies Suzy's throw (ST) as the cause of the bottle shattering (BS) by assuming that stopping Suzy's throw and preventing Billy from hitting would result in the bottle not shattering. However, this reasoning is counter-intuitive and violates the structural model. If Suzy does not throw but Billy does, Billy must hit the bottle, making the assumption that Billy does not hit inconsistent. This issue arises from selectively applying interventions which leads to impossible scenarios.

While this approach provided important computational insights into causal models, it also has several limitations. A significant drawback of structural-equations modeling is its inability to distinguish between conditions and transitions, making it challenging to differentiate between enduring conditions (e.g., a bridge is collapsed) and transitional events (e.g., a bridge collapses). For example, consider a bridge that collapses due to an earthquake but would have been demolished for construction five minutes later. While the earthquake is the immediate cause of the collapse, it may not be accurate to attribute the bridge's state of being collapsed far into the future solely to the earthquake. This distinction is crucial for accurate causal determinations. Additionally, structural-equations modeling struggles with distinguishing between the presence and absence of events, particularly action versus inaction. Also, fluents cannot change after becoming true, and all events are assumed to be independent and have no preconditions. Finally, temporal ordering of events is also ignored. These limitations highlight the need for more sophisticated frameworks to address these nuances for precise causal reasoning.

2.6 Causality and Counterfactuals in the Situation Calculus

Structural equation models struggle to differentiate between the presence and absence of events, a challenge that the situation calculus specification overcomes by distinguishing between actions and enduring conditions. For instance, it can differentiate between turning off a light (an action) and the room being dark (an enduring

condition) by incorporating fluents. Hopkins and Pearl [26] propose causal models within the framework of the situation calculus, which are sequences of possible or necessary actions while omitting those that are not. These actions represent real events and allow for evaluating counterfactual scenarios, offering a more flexible and expressive approach to reasoning about causation than structural-equations models.

Given a situation S which is a sequence of actions $(a_1, a_2, \ldots, a_n)$, and a set of actions A as a subset of actions in S, the expression $S - A$ denotes the situation that results from removing all actions in A from S. A potential situation is represented as a pair (S, F), where S is the sequence of actions, and F is a function that specifies which actions in S should be considered or ignored. A situation calculus causal model M comprises a setup D and a potential situation P. D represents situation calculus basic action theory axioms (3.2.2) that define both domain-dependent and domain-independent aspects of actions and their effects on properties and situations. The model adjusts S and F to determine which actions are necessary or should be omitted. The natural executable substitution $NES(M)$ for the model executes each action in P if feasible, demonstrating the actual sequence of events under the model. After establishing hypothetical scenarios within a situation calculus specification D, one can ask basic questions like whether something is true, if something has a specific value, or if a particular action happened. In this framework, a statement takes the form $[\neg a_1, \ldots, \neg a_m, b_1, \ldots, b_n]q(U)$, where a_1 to a_m are actions that are omitted, b_1 to b_n are actions that are executed in the hypothetical scenario, q represents the query, and U describes the initial situation S_0. To determine the truth of such a statement

in a causal model $M = (D, P)$, one verifies if D and U support it, reflecting the actual events.

This approach advocates using situation calculus for causality due to its expressivity, but it lacks clear guidelines for selecting causal models and does not address action preconditions or methods for identifying actual causes/actions responsible for a condition.

2.7 Bochman's non-monotonic Account

Bochman [5] argues that previous models relied heavily on counterfactual reasoning or regularity accounts without sufficiently addressing the nuances of action and change in dynamic systems. Motivated by the need for a more expressive and formalized approach, Bochman [5] proposed a new definition of actual achievement causes in the non-monotonic framework of causal calculus introduced by McCain and Turner [39]. This definition is based on the NESS (Necessary Element of a Sufficient Set) condition [57], identifying an actual cause as an action or event that is essential to a condition necessary and sufficient for an effect to occur. The study does not address two major representational issues: the use of multi-valued variables and the role of normality and defaults in reasoning about actual causation.

2.8 Batusov and Soutchanski's Foundational Account of Causes

Inspired by the work of Halpern and Pearl [22] and addressing the limitations of structural equations, Batusov and Soutchanski [4] proposed a solution for actual causality within the more expressive framework of the situation calculus. Their approach requires that the effect must be achieved within the scenario for its achievement causes to be computed. They provide definitions for both direct (primary) and indirect (secondary) causes. A direct cause is an action that directly brings about the effect, while an indirect cause either enables the execution of the primary cause or makes the condition of interest conditionally or partially true. Additionally, Batusov and Soutchanski introduce the concept of maintenance causes, which are responsible for sustaining the achievement of the effect. To compute the entire chain of actions responsible for the effect, they utilize a regression operator introduced by Reiter [52]. However, their method is limited to scenarios that are linear and involve only discrete changes. I will discuss their approach further in Chapter 3 as it serves as a foundation for my research.

2.9 Causal Knowledge and its Dynamics

Khan and Lespérance [30] argued that causality should be considered not only from an objective standpoint but also from the agent's perspective. They noted that in situations where knowledge is incomplete, an agent may be unable to determine

the cause of an effect. However, in dynamic domains, an agent can acquire more knowledge through *sensing* actions or when another agent informs her about certain facts. To address these issues, Khan and Lespérance, building on the work of Batusov and Soutchanski [4], proposed an inductive definition of actual causes, essentially embedding the formalization of actual causes into the language of the situation calculus. This is in contrast to Batusov and Soutchanski's work which is meta-theoretical formalization of causes. This allowed them to naturally combine the notions of cause and knowledge and study the epistemics of causation and causal knowledge dynamics. They demonstrate that enabling such introspective behavior in agents is beneficial in multi-agent systems, allowing agents to interact, collaborate, identify the agent responsible for an effect, and prevent other agents from performing certain actions, thus enriching the domain dynamics. However, like the earlier approach, their framework handles only discrete changes and linear scenarios. I will use their definition to define direct causes in hybrid domains. I will discuss their approach in detail in Chapter 3.

2.10 Necessary and Sufficient Conditions for Actual Root Causes

While the aforementioned work on causation appeals to intuition, some argue for a counterfactual perspective. Khan and Soutchanski [32], aligning with the counterfactual approach, defined necessary and sufficient conditions for achievement causes.

They define necessary causes as actions contributing to an effect without being preempted, and whose effects persist until the scenario's end. Sufficient causes are actions that must bring about the effect upon execution. Additionally, they define counterfactual dependence of an action, indicating the effect would not occur without it. They introduce "enduring producers" as actions that meet both necessary and sufficient conditions and demonstrate that their definition aligns with Batusov and Soutchanski's earlier work [4]. They discussed how this counterfactual account of causation can also be interpreted from a regularity perspective related to INUS condition. This work thus contributes to the ongoing debate between the counterfactual and regularity approaches by combining both perspectives, offering a comprehensive analysis that supports a more unified understanding of causation.

2.11 Anil Nerode's Hybrid Systems

Anil Nerode [43] introduced the term "Hybrid Systems" to describe systems where discrete and continuous processes interact in real-time. These systems typically consist of plants, whose evolution is to be constrained, sensors, which transfer plant state measurements to controllers, controllers, which issue control orders, and actuators, which change the plant state. The challenge lies in designing sensors and controllers to meet specific system requirements, which is particularly important in fields like air traffic control, supply chain management, and network routing. In the 1980s, hybrid

systems theory was developed to systematically address these challenges by converting discrete constraints into continuous ones, enabling the transformation of complex systems into continuous models with approximate optimal controls achieved through finite control automata [33].

The literature on hybrid systems has grown significantly, particularly in verification problems. Nerode's contributions to linear control, along with Wolf Kohn's Declarative Control approach using PROLOG [43] at Boeing, have played a key role in this expansion. Conferences, such as the 1992 International Hybrid Systems Conference at Cornell, have fostered collaboration and established a common framework for hybrid systems research. Despite these advancements, ensuring the robustness and accuracy of control models remains critical, with extensive testing needed to address potential inaccuracies. Future research will likely explore deeper connections between hybrid systems and logical frameworks to drive further progress in causation and control [33].

2.12 Causal Models with Infinitely Many Variables

Traditional structural-equations models face limitations when dealing with complex systems, especially those with infinitely many variables or continuous ranges. Recently, Perters and Halpern [23] introduced Generalized Structural-Equations Models that extend the capabilities of traditional structural-equations models to represent complex models like systems of differential equations and hybrid automata, allowing

for more flexible and accurate modeling of causal relationships and interventions.

Formally, a generalized structural-equations model $\mathcal{M}$ is a pair $(\mathcal{S}, \mathcal{F})$, where $\mathcal{S}$ is a signature, and $\mathcal{F}$ is a mapping from contexts and interventions to sets of outcomes. The signature is the same, a quadruple $(\mathcal{U}, \mathcal{V}, \mathcal{R}, \mathcal{I})$, defines variables, their possible values, and allowed interventions. Unlike in structural-equations models, $\mathcal{U}$ and $\mathcal{V}$ are not required to be finite, nor are the possible outcomes mapped by the function $\mathcal{R}(Y)$. In generalized structural-equations models, the number of variables ($\mathcal{U}$ and $\mathcal{V}$) and their ranges $\mathcal{R}(Y)$ can be infinite. Given an intervention and context, generalized structural-equations models produce a set of variable assignments (outcomes) that can be infinite, whereas outcomes in structural-equations models are limited due to the finite number of variables. It can represent variables indexed by time or other continuous parameters. Additionally, generalized structural-equations models allow specifying which interventions are permitted, providing a more expressive feature than structural-equations models.

Consider an example of managing traffic flow at an intersection. To model the effects of changing a traffic light's state on vehicle behavior, consider variables like the number of cars arriving at the intersection per second (C_t), the state of the traffic light (L_t) with possible values $\{green, yellow, red\}$, and the drivers' reaction times (R_t). An intervention at $t = 5$ changes the light to red ($L_t \leftarrow red$). The outcomes depend on contextual factors like weather, traffic density, and driver behavior. Some drivers may stop immediately ($C_t \to 0$), others may slow down gradually, and adverse weather might delay stopping further ($C_t > 0$ for a longer period). The generalized

structural-equations models map the intervention and context to a set of possible outcomes for C_t, representing different traffic patterns post-intervention. By accommodating multiple possible results, generalized structural equation models provide a more comprehensive framework for analyzing such dynamic and uncertain scenarios.

Despite their increased expressiveness, generalized structural-equations models face several challenges. The potential for infinitely many outcomes complicates reasoning and analysis, often necessitating a restriction to a language that references only countably many values and interventions explicitly. Additionally, notable issues persist: fluents cannot change once they become true, the models do not handle dependencies between events (however, actions without preconditions is counter-intuitive), and they do not account for the temporal order of event occurrences. Moreover, the lack of a directed acyclic graph (DAG) structure in generalized structural-equations models limits the use of graphical tools like the do-calculus, which are widely used in structural-equations modeling analysis. While this trade-off limits certain analytical capabilities, generalized structural-equations models offer broader applicability and expressiveness, particularly for representing dynamical systems and complex causal relationships that structural-equations models cannot handle.

2.13 Conclusion

In this chapter, I outlined some previous work on actual causation, hybrid dynamic frameworks, and causal models that can support infinitely many variables. In the

next chapter, I will give details of some of these works, in particular of those that are directly related to my proposal.

Chapter 3

Foundations

3.1 Introduction

In this chapter, I discuss previous work on formalizing dynamic domains, which will form the basis of my work in the next chapters. I start with our base framework, the situation calculus, which is a dialect of first-order logic that can be used to reason about actions and their effects. I then discuss a variant of the situation calculus, namely Hybrid Temporal Situation Calculus, that supports domains with both discrete and continuous change. I use this in Chapter 4 as my base framework for defining primary cause in hybrid dynamic domains. I also discuss previous work on formalizing achievement causes in discrete domains within the situation calculus. Finally, I propose a specification of counterfactual worlds in discrete domains, which I will later extend to handle hybrid domains in Chapter 5. This part is a new contribution.

3.2 The Situation Calculus

3.2.1 Introduction

The situation calculus is a well-known second-order language for representing and reasoning about dynamic worlds. The formalism was first introduced by John McCarthy [40], and later it was axiomatized and further developed by Raymond Reiter [52]. In the situation calculus, all changes in the world result from named action terms. Thus, actions are first-order terms; for example, $pickUp(r, d)$ might indicate the robot r's action of picking up device d, $activate(r, d)$ might mean r's action of activating device d, and $move(r, a, b)$ can be used to denote r moving from location a to location b, etc.

Situations represent a possible world history resulting from performing some actions. In the situation calculus, a model of the possible evolution of the world forms a tree-like structure, where each node corresponds to a situation, starting from the initial situation and branching out with each action execution. Each branch consists of a sequence of actions occurred in some possible evolution and their resulting situations. The constant S_0 is used to denote the initial situation where no action has been performed yet, meaning S_0 has no predecessor situation. There is a distinguished binary function symbol do; $do(a, s)$ denotes the successor situation to s resulting from performing the action a. For example, $do(activate(r, d), s)$ denotes the successor situation resulting from r activating device d when the current situation is

s. The expression $do([a_1, \cdots, a_n], s)$ is an abbreviation that can be defined as:

$$do([a_1, \cdots, a_n], s) \doteq do(a_n, do(a_{n-1}, \cdots do(a_1, s) \cdots)). \tag{3.1}$$

It thus represents the situation resulting from executing actions $a_1, \cdots, a_n$, starting in situation s.

There is a special predicate $Poss(a, s)$, which indicates that action a is executable in situation s . Additionally, the binary predicate $s \sqsubset s'$ denotes that situation s' can be reached from s by executing some sequence of actions. $s \sqsubseteq s'$ is defined as:

$$s \sqsubseteq s' \equiv s \sqsubset s' \vee s = s'. \tag{3.2}$$

Moreover, $s < s'$ is an abbreviation of $s \sqsubset s' \wedge Executable(s')$, where $Executable(s)$ is defined as follows.

Definition 3.2.1 (Executable Situation)**.**

$$Executable(s) \stackrel{\text{def}}{=} \forall a, s'. do(a, s') \sqsubseteq s \supset Poss(a, s').$$

That is, a situation s is executable if and only if all actions in its history were possible to execute in the order of their occurrence.

Fluents in the situation calculus are situation-dependent properties of interest that change their truth values due to action executions. Relational fluents, denoted

by predicate symbols with a situation term as the last argument, have truth values that vary between situations. For example,"$Holding(r, d, s)$" might indicate that a robot r is holding a device d in situation s. Functional fluents, denoted by function symbols with a situation term as the last argument, have values that vary between situations. For instance,"$location(d, s)$" can be used to return the location of device d in situation s.

We will also use a notion of uniform formula in the situation calculus.

Definition 3.2.2 (Uniform Formula in σ)**.** *A formula is uniform in situation σ if and only if it does not mention the predicates Poss or $\sqsubset$, it does not quantify over variables of sort situation, it does not mention equality on situations, and whenever it mentions a term of sort situation in the situation argument position of a fluent, then that term is σ.*

3.2.2 Basic Action Theory

An action theory refers to a collection of axioms that describe how the state of the world changes as a consequence of action execution. In the situation calculus, a dynamic domain is specified using a *basic action theory (BAT)* that includes the following sets of axioms:

1. First-order initial state axioms $\mathcal{D}_{S_0}$, which indicates what is true initially.

2. First-order action precondition axioms $\mathcal{D}_{ap}$, characterizing $Poss(a, s)$.

3. First-order successor-state axioms $\mathcal{D}_{ss}$, indicating precisely when and how the fluents change.

4. First-order unique names axioms $\mathcal{D}_{una}$ for actions, stating that different action terms represent distinct actions.

5. Second-order domain-independent foundational axioms Σ, describing the structure of situations.

Overall, the situation calculus basic action theory $\mathcal{D}$ is a collection of axioms, $\mathcal{D} \doteq \Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0}$ [35, 52]. I now explain each of these types of axioms.

Initial State Axioms

These are used to specify the initial values of the fluents at the start of the reasoning process, i.e., in S_0. The actual content of the initial state axioms will depend on the specific domain being modeled. For example, one might use the following axiom to state that a device D_1 is not active in situation S_0:

Axiom 3.2.1.

$$\neg Active(D_1, S_0).$$

Action Precondition Axioms

Actions often have preconditions that must be satisfied in a specific situation before they can be performed in that situation. Recall that the predicate $Poss(a, s)$ states that action a is physically executable in situation s. In situation calculus basic

action theory, every action is associated with a precondition axiom of the following form, where $\Pi_{Poss}(a, s)$ is a formula that is uniform in situation s[1].

Axiom 3.2.2.

$$Poss(a, s) \equiv \Pi_{Poss}(a, s).$$

For example, if for a robot r it is possible to tag a new price d to some product p in situation s if and only if r is next to p, and the current price (modeled using the function *price*) is different from the new price (d), then this can be specified using the following axiom.

Axiom 3.2.3.

$$Poss(tagPrice(r, p, d), s) \equiv NextTo(r, p, s) \wedge \neg(d = price(p, s)).$$

Successor-State Axioms

Before we discuss successor-state axioms, let us reflect on the motivation behind adding these.

Effect Axioms. Any action theory must encode the evolution of the world brought about by action execution. In the situation calculus, such changes in the world can be specified using effect axioms, which describe how actions affect fluents. For relational fluents, there can be positive effect axioms, stating when a fluent changes its value from true to false, and negative effect axioms, describing actions

[1]Henceforth, all free variables are assumed to be universally quantified from the outside, and thus, e.g., $P(x, y) \equiv Q(x, y)$ stands for $\forall x, y.P(x, y) \equiv Q(x, y)$.

and conditions under which fluent changes from false to true. For instance, a positive effect axiom for the relational fluent $SoldOut(p, s)$ can be specified as follows.

Axiom 3.2.4.

$$inStock(p, s) = 1 \supset SoldOut(p, do(sell(p), s)).$$

That is, product p is sold out after someone sells the last stock of p in situation s. Similarly, a negative effect axiom for $SoldOut(p, s)$ can be specified as follows.

Axiom 3.2.5.

$$\neg SoldOut(p, do(restock(p, n), s)).$$

Thus, p is no longer sold out after n number of it is restocked in situation s. For functional fluents, we only need one effect axiom.

Frame Axioms. The problem with effect axioms is that these do not specify which fluents remain unchanged when an action is performed. For example, *pickUp* and *drop* actions do not change the *price, availability, color, weight, category,* etc., of a product. Thus, to specify that picking up a product does not change its weight, one can use the following axiom:

Axiom 3.2.6.

$$weight(p, s) = x \supset weight(p, do(pickUp(r, p), s)) = x.$$

In general, there are of the order of $2 \times A \times F$ frame axioms, where A is the number of actions and F is the number of fluents. The frame problem [41] involves

the specification of this overwhelming number of frame axioms, as identifying all the fluents that don't change when an action is executed can be just too big of a task. Different proposals have been made to address the frame problem. Pednault's proposal [47] suggests systematically generating frame axioms from effect axioms, though it does not reduce the number of frame axioms needed. The proposal by Davis [7], Haas [17], and Schubert [54] introduces explanation closure axioms, which offer a more compact representation by universally quantifying over actions. Explanation closure axioms make a causal completeness assumption, stating that a fluent can change its value only under the specified conditions in the effect axioms. This means that if a fluent changes its value from false to true, the positive effect axiom must have been true, and if it changes from true to false, the negative effect axiom must have been true. This reduces the number of required axioms from $2 \times A \times F$ to $2 \times F$. However, Schubert [54] argues that explanation closure axioms cannot be systematically derived from effect axioms and must be independently provided by the axiomatizer.

Building on these ideas, Reiter [52] developed successor-state axioms by combining effect axioms with the causal completeness assumption and introducing a consistency assumption, which states that the conditions under which a fluent becomes true when an action is executed in some situation and those under which it becomes false are never jointly satisfied. The successor-state axiom for each fluent states how exactly a fluent changes its value as a result of actions, and sufficiently encodes both effect and frame axioms when the causal completeness and consistency assumptions hold.

Each fluent is associated with a single successor-state axiom. While these can be automatically generated given the effect axioms, one can also specify these directly (and get rid of the effect axioms). Each of these has the following form.

Axiom 3.2.7.

$$F(\overline{x}, do(a, s)) \equiv \Phi_F(\overline{x}, a, s),$$

where, $\Phi_F(\overline{x}, a, s)$ is a formula that is uniform in s, and typically has the following form:

$$\Phi_F^+(\overline{x}, a, s) \vee (F(\overline{x}, s) \wedge \neg\Phi_F^-(\overline{x}, a, s)).$$

Here, $\Phi_F^+(\overline{x}, a, s)$ (and $\Phi_F^-(\overline{x}, a, s)$) specify when fluent F changes to true (and false, respectively). Successor-state axioms are more complex than effect axioms but are much fewer in number (one per fluent) compared to total effect and frame axioms. For example, *SoldOut* might have the following successor-state axiom:

Axiom 3.2.8.

$$(inStock(p, s) = 1 \wedge SoldOut(p, do(a, s)) \equiv a = sell(p))$$
$$\vee (SoldOut(p, s) \wedge \neg\exists n.(a = restock(p, n))).$$

That is, p is sold out after action a has been performed in situation s, i.e., in $do(a, s)$, if and only if a refers to selling the last stock of p, or if p was already sold out in s and a is not the action of restocking some amount n of p.

Unique Names Axioms

Successor-state axioms rely on the commonsense assumption, that different action terms represent different actions. For example, we might want to state that action *pickup* is not the same as action *drop*, i.e., $pickup(x) \neq drop(x)$. In general, for two distinct action names f_1 and f_2, we have:

Axiom 3.2.9.

$$f_1(\overline{x}) \neq f_2(\overline{y}).$$

Also identical actions have identical arguments:

Axiom 3.2.10.

$$f(\overline{x}) = f(\overline{y}) \supset \overline{x} = \overline{y}.$$

All the axioms that I discussed above are domain-dependent ones. We also need the following domain-independent foundational axioms.

Foundational Axioms

Foundational axioms define the structure of situations and the properties of *do* and $\sqsubset$. These include the following four axioms.

1. *do* is injective:

$$do(a_1, s_1) = do(a_2, s_2) \supset a_1 = a_2 \wedge s_1 = s_2. \tag{3.3}$$

2. There are no situations other than those reachable from S_0:

$$\forall P.\ P(S_0) \wedge \forall a, s.\ [P(s) \supset P(do(a, s))] \supset \forall s.\ P(s). \tag{3.4}$$

3. The following axioms specify $\sqsubset$ as subhistory:

$$\neg s \sqsubset S_0, \tag{3.5}$$

$$s \sqsubset do(a, s') \equiv s \sqsubseteq s'. \tag{3.6}$$

The following are some basic logical consequences of the foundational axioms [41]:

$$S_0 \neq do(a, s) \tag{3.7}$$

$$do(a, s) \neq s \tag{3.8}$$

Existence of a predecessor: $s = S_0 \vee \exists a, s'.\ s = do(a, s')$ (3.9)

Grounded in S_0 : $S_0 \sqsubseteq s$ (3.10)

Transitivity: $s_1 \sqsubset s_2 \wedge s_2 \sqsubset s_3 \supset s_1 \sqsubset s_3$ (3.11)

Anti-reflexivity: $\neg(s \sqsubset s)$ (3.12)

Unique names: $s_1 \sqsubset s_2 \supset s_1 \neq s_2$ (3.13)

Anti-symmetry: $s \sqsubset s' \supset \neg(s' \sqsubset s)$ (3.14)

$$\neg(do(a, s) \sqsubseteq s) \tag{3.15}$$

$$s \sqsubseteq s' \wedge s' \sqsubseteq s \supset s = s' \tag{3.16}$$

3.2.3 Reasoning in the Situation Calculus

One way to view reasoning is as the process of inferring new knowledge, those that are logical consequences of the explicitly stated knowledge and action specification in the knowledge base. Put otherwise, it is a mechanism by which implicit knowledge is made explicit. A particularly useful type of reasoning in a dynamic setting like the situation calculus involves determining if a formula φ is true after a sequence of actions $a_1, a_2, \cdots, a_n$ has been performed, starting in the initial situation S_0, i.e., whether $\varphi[do([a_1, a_2, \cdots, a_n], S_0)]$. A special case of this is to check whether the sequence is executable, i.e., $\mathit{Executable}(do([a_1, a_2, \cdots, a_n], S_0)])$. These problems are instances of the projection problem. The situation calculus provides two techniques for solving the projection problem, progression, which involves progressing the knowledge base $\mathcal{D}$ with the given sequence of actions and then checking whether φ holds in the progressed theory (which we will not discuss further here), and goal regression, which is detailed below.

Regression

Goal regression is a backward inference process. It involves regressing the goal φ (instead of the theory $\mathcal{D}$) and finding the minimal preconditions that must be true initially for the given action sequence to achieve φ. Then we can simply check the regressed version of φ relative to the initial theory $\mathcal{D}_{S_0}$.

The main idea behind regression is that since the right-hand side of the successor-state axioms and action precondition axioms are carefully designed to be uniform in s (thus preserving the Markovian property), if the number of actions in $do(a, s)$ is known, one can always replace a fluent $F(do(a, s))$ with the right-hand side of the successor-state axiom for F, which only mentions s. Similarly, if the action function A is known, one can always replace $Poss(A, s)$ with the right-hand side of the action precondition axiom for A, which only mentions s. By repeatedly applying these two rules, one can syntactically transform the formula $\varphi[do([a_1, a_2, \cdots, a_n], S_0)]$ to a formula φ' that only mentions S_0. Thus the task of evaluating $\varphi[do([a_1, a_2, \cdots, a_n], S_0)]$ relative to $\mathcal{D}$ becomes that of evaluating φ' relative to the initial theory. In other words, regression of φ with respect to $do([a_1, a_2, \cdots, a_n], S_0)$ is the weakest precondition φ' that must be true in the initial situation S_0 for the sequence $[a_1, a_2, \cdots, a_n]$ to bring about the φ.

A key feature of basic action theories is the existence of a sound and complete *regression mechanism* for answering queries about situations resulting from performing a sequence of actions [48, 52]. In a nutshell, the regression operator $\mathcal{R}^*$ reduces a formula ϕ about a particular future situation to an equivalent formula $\mathcal{R}^*[\phi]$ about the initial situation S_0.

A formula ϕ is regressable if and only if (i) all situation terms in it are of the form $do([a_1, \ldots, a_n], S_0)$, (ii) in every atom of the form $Poss(a, \sigma)$, the action function is specified, i.e., a is of the form $A(t_1, \ldots, t_n)$, (iii) it does not quantify over situations, and (iv) it does not contain $\sqsubset$ or equality over situation terms. Thus in essence, a

formula is regressable if it does not contain situation variables.

In the following, I give a one-step variant of $\mathcal{R}^*$, $\mathcal{R}$.

Definition 3.2.3 (The Regression Operator)**.** *[50]*

(1) When ϕ is a non-fluent atom, including equality atoms without functional fluents as arguments, or when ϕ is a fluent atom, whose situation argument is S_0, $\mathcal{R}[\phi] = \phi$.

(2a) For a non-functional fluent F, whose successor-state axiom in $\mathcal{D}$ is $F(\vec{x}, do(a, s)) \equiv \Phi_F(\vec{x}, a, s)$, $\mathcal{R}[F(\vec{t}, do(\alpha, \sigma))] = \Phi_F(\vec{t}, \alpha, \sigma)$.

(2b) For an equality literal with a functional fluent f, whose successor-state axiom is $f(\vec{x}, do(a, s)) = y \equiv \Phi_f(\vec{x}, y, a, s)$, $\mathcal{R}[f(\vec{t}, do(\alpha, \sigma)) = t'] = \Phi_f(\vec{t}, t', \alpha, \sigma)$.

(2c) For a Poss literal with the action precondition axiom of the form $Poss(A(\vec{x}), s) \equiv \Pi_A(\vec{x}, s)$, $\mathcal{R}[Poss(A(\vec{t}), \sigma)] \equiv \mathcal{R}[\Pi_A(\vec{t}, \sigma)]$.

(3) For any non-atomic formulae, regression is defined inductively: $\mathcal{R}[\neg\phi] = \neg\mathcal{R}[\phi]$, $\mathcal{R}[\phi_1 \wedge \phi_2] = \mathcal{R}[\phi_1] \wedge \mathcal{R}[\phi_2]$, $\mathcal{R}[\exists v.\ \phi] = \exists v.\ \mathcal{R}[\phi]$.

$\mathcal{R}^*$ can then be defined as the repeated application of $\mathcal{R}$ until further applications leave the formula unchanged. These results have facilitated the implementation of integrity constraints in databases, planning tasks, and model checking in high-level agent programming languages like Golog [36] and ConGolog [8].

3.2.4 Limitations of the Situation Calculus

While situation calculus has been widely used to formalize and study various aspects of dynamic domains and artificial intelligence, the basic variant does come

with its own limitations. However, various researchers have addressed most of these issues by proposing variants of the situation calculus. I will mention the following:

- In the situation calculus, all changes are due to named actions; as well, it primarily deals with discrete change, and cannot represent continuous change. See [1] for an account that deals with continuous change and supports change due to the passage of time.

- Here, action effects are deterministic. See [10] for a recent proposal that deals with non-deterministic actions.

- It focuses on qualitative representations of information rather than quantitative. See [49, 6] that incorporates decision theoretic aspects in the situation calculus.

- It does not allow concurrent actions and events. See [51] for an extension that models this, and [15] for a formalization of multiplayer synchronous games in the situation calculus.

- It does not take an intentional stance towards agent goals. See, e.g., [53] for a formalization of knowledge and knowledge change, [12] for a model of iterated belief change, [28] for an account of intentions and its dynamics, [13] for a theory of conditions and joint ability, and [29] for belief-desite-intentions (BDI) agent programming language, all based on the situation calculus.

- It does not handle planning on high-level agent programming. Refer to the languages in the Golog family of the high-level programming languages, such

as Golog [36], ConGolog [8], IndiGolog [9], etc. for solutions.

For this thesis, I am particularly interested in hybrid temporal situation calculus, which extends situation calculus to hybrid domains. I will thus discuss this in detail in the next section.

3.3 Hybrid Temporal Situation Calculus

3.3.1 Introduction

The situation calculus only deals with discrete actions and abrupt change of fluents due to these actions, and does not incorporate a standard notion of time; but in the real world, change can be continuous as well as discrete. For example, a change in room temperature after adjusting the thermostat and a change in weather conditions happen over time, but not immediately. To deal with time and continuous change in the framework of situation calculus, Reiter proposed temporal situation calculus [52]. The intuition behind Reiter's idea is that all changes in the world, no matter whether discrete or continuous, are results of discrete actions. For each continuous process, there is an action that initiates the change, and there is another instantaneous action that terminates the change. To accommodate time, Reiter introduced two special functions, $time(a)$, which refers to the time at which an action a is executed, and $start(s)$, which refers to the starting time of the situation s. He requires every action $a(\vec{x}, s)$ to take a time argument t. *time* of an action a is specified using the following axiom:

Axiom 3.3.1.

$$time(a(\vec{x}, t)) = t.$$

It is included in $\mathcal{D}_{S_0}$ for every action function $a(\vec{x}, t)$ in the domain. Similarly, the function *start* is specified by a new foundational axiom:

Axiom 3.3.2.

$$start(do(a, s)) = time(a).$$

The starting time of S_0 is not enforced.

Now, consider a situation $do(a(x, 1),\ do(b(x, 2), S_0))$, where the time of action a is before the time of action b, but action a is executed after b. This leads to logical inconsistencies. To prevent such temporal paradoxes, executable situations are redefined as follows:

Definition 3.3.1 (Executable Temporal Situation)**.**

$$Executable(s) \stackrel{\text{def}}{=} \forall a, s'.\ do(a, s') \sqsubseteq s \supset (Poss(a, s') \wedge start(s') \leq time(a)).$$

That is, a situation s is executable if and only if all actions in the sequence leading to s were possible to execute in the order of their occurrence, and the time of each action cannot be earlier than that of the situation where it was performed. Note that this allows multiple actions to be executed at the same time.

In Reiter's temporal situation calculus [52], fluents remain atemporal and do

not actually change with time; instead, they attain certain values when some time-stamped actions are performed. The limitation here is that one cannot query the value of a continuous fluent at any arbitrary time. For example, in a changing room temperature scenario, the room's current temperature cannot be determined at a specific moment without referencing a time-stamped action. Soutchanski [56] suggested that one could use an auxiliary exogenous action $watch(t)$ which fixes a time point t to a situation when it occurs, allowing one to pose an atemporal query in the resulting situation. Similarly, an auxiliary exogenous action $waitFor(\phi)$ can be introduced, which is executed when the condition ϕ becomes true.

Recently, Batusov and Soutchanski proposed the hybrid temporal situation calculus [1, 2], which draws inspiration from Reiter's work described above and hybrid systems in control theory [43]. This approach is based on discrete transitions between states that continuously evolve over time. The latter change is dictated by discrete contexts brought about by actions. For example, in a weather prediction system, the temperature changes continuously depending on whether it is sunny or not, but the state change from being sunny to cloudy can be viewed as discrete.

The hybrid temporal situation calculus reuses Reiter's basic action theory (see Section 3.2.2), with the functions $time(a)$ and $start(s)$ defined as above. Additionally, in hybrid temporal situation calculus, the situation calculus (atemporal) fluents are preserved. These atemporal fluents no longer represent continuous change but rather provide a *context* within which the values of temporal fluents can change. For instance, the position of a car, a continuous fluent, changes over time when the car

is in the state of actively being driven. Here, the discrete fluent $Driving(p, c, s)$, i.e., a person p is driving car c in situation s, serves as a context that influences how the continuous functional fluent $position(c, t, s)$, representing the position of car c at time t in situation s, varies with time.

To model continuous change, in addition to Reiter's successor-state axioms, hybrid temporal situation calculus includes state evolution axioms [1, 2], each of which defines how a temporal fluent's value changes over time when some relevant context is enabled.

3.3.2 State Evolution Axioms

The foundation of state evolution axioms is Reiter's temporal change axioms [52]. These axioms describe the change in the value of each temporal fluent in an arbitrary situation over time and have the general form of:

$$\gamma(\overline{x}, s) \wedge \delta(\overline{x}, y, t, s) \supset f(\overline{x}, t, s) = y, \tag{3.17}$$

where, $t, s, \overline{x}$, and y are variables, and $\gamma(\overline{x}, s)$ is the context that specifies the conditions under which the formula $\delta(\overline{x}, y, t, s)$ is to be used to compute the value of fluent $f(\vec{x})$ at time t. There is no restriction on the formula δ, and it can be algebraic, logical, or differential equations to compute values appropriately. The value of a temporal fluent can change differently under different contexts. The set of k temporal change axioms for fluent $f(\vec{x})$ can be expressed as:

Axiom 3.3.3.

$$\Phi(\overline{x}, y, t, s) \supset f(\overline{x}, t, s) = y,$$

where $\Phi(\overline{x}, y, t, s)$ is $\bigvee_{1 \leq i \leq k}(\gamma_i(\overline{x}, s) \wedge \delta_i(\overline{x}, y, t, s))$, and it states that formula δ_i can be used to determine the change in f if the context δ_i is activated. In hybrid temporal situation calculus, the contexts are mutually exclusive to make sure continuous fluents do not assume two different values at the same time. To this end, the following condition is added to the background theory:

Axiom 3.3.4.

$$\Phi(\overline{x}, y, t, s) \wedge \Phi(\overline{x}, y', t, s) \supset y = y'.$$

Finally, a causal completeness assumption is required to state that a temporal fluent's value cannot change if no context is true [1, 2]. This is formally stated as:

Axiom 3.3.5.

$$f(\overline{x}, t, s) \neq f(\overline{x}, start(s), s) \supset \exists y.\Phi(\overline{x}, y, t, s).$$

Let $\Psi(\overline{x}, s)$ denote $\bigvee_{1 \leq i \leq k} \gamma_i(\overline{x}, s)$, which is a set of all the contexts related to a temporal fluent. By combining the temporal change axiom (Axiom 3.3.3) with the causal completeness assumption axiom (Axiom 3.3.5), state evolution axioms are obtained:

Axiom 3.3.6 (State Evolution Axiom)**.**

$$f(\overline{x}, t, s) = y \equiv [\Phi(\overline{x}, y, t, s) \vee y = f(\overline{x}, start(s), s) \wedge \neg\Psi(\overline{x}, s)].$$

Thus the above state evolution axiom states that the value of a temporal fluent $f(\overline{x})$ changes only if some context γ_i holds and according to the rules defined in the formula δ_i associated with γ_i; otherwise it remains the same as at the beginning of the situation. The formula $\delta_i(\vec{x}, y, t, s)$ implicitly or explicitly defines y using some arbitrary (domain-specific) constraints on variables and fluents.

State evolution axioms only talk about the continuous change in the value of a temporal fluent in an arbitrary situation, but it does not fully capture the hybrid phenomenon. We need a successor-state axiom for each temporal fluent to represent action-induced discrete changes in fluents. To capture this hybrid phenomenon, for each temporal functional fluent $f(\overline{x}, t, s)$, an additional auxiliary atemporal functional fluent $f_{init}(\overline{x}, \mathrm{s})$ along with its successor-state axiom is introduced.

We also need the following condition:

Axiom 3.3.7.

$$\neg\exists y(e(\overline{x}, y, a, s)) \supset f_{init}(\overline{x}, do(a, s)) = f(\overline{x}, time(a), s).$$

It assumes that the effect axiom from which the successor-state axiom for f_{init} was defined has the form $e(\overline{x}, y, a, s) \supset f_{init}(\overline{x}, do(a, s)) = y$ and states that if no relevant effect axiom is invoked by an action a then f_{init} takes the most recent value of the associated temporal fluent $f(\overline{x}, t, s)$. To keep consistency between temporal fluents and their relevant atemporal fluents, it is required that in an arbitrary situation, the continuous evolution of each temporal fluent f starts with the value computed for

f_{init} by its successor-state axiom.

Hybrid Temporal Situation Calculus Basic Action Theory

A *hybrid basic action theory* [1] then is a collection of axioms:

Axiom 3.3.8.

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{se},$$

where, the following are defined the same as in the situation calculus basic action theory (see Section 3.2.2): Σ (foundational axioms), $\mathcal{D}_{ss}$ (successor-state axioms), $\mathcal{D}_{ap}$ (action precondition axioms), $\mathcal{D}_{una}$ (unique names axioms), and $\mathcal{D}_{S_0}$ (initial-state axioms). And, $\mathcal{D}_{se}$ is the set of state evolution axioms. Also, as discussed above, hybrid basic action theory requires that every action mentioned in $\mathcal{D}$ is temporal and has a time argument.

To maintain consistency within the framework, we need to define the notion of *stratified basic action theory.* First, let me state the stratified state evolution axiom.

Definition 3.3.2 (Stratified State Evolution Axiom)**.** *A set of state evolution axioms D_{se} is stratified if and only if there are no temporal fluents f_1, ..., f_n such that $f_1 \succ f_2 \succ ... \succ f_n \succ f_1$, where $f \succ f'$ holds if there is a state evolution axiom in $\mathcal{D}_{se}$, where f appears on the left-hand side and f' on the right-hand side.*

Using this, stratified temporal basic action theories are defined as follows.

Definition 3.3.3 (Stratified Temporal Basic Action Theory)**.** *A temporal basic action theory $\mathcal{D}$ is stratified if and only if its state evolution axioms $\mathcal{D}_{se}$ are stratified.*

Simply put, the stratification condition implies that there should be no cyclical relationships among the temporal fluents. It can be shown that a stratified temporal basic action theory $\mathcal{D}$ is satisfiable if $\mathcal{D}_{una} \cup \mathcal{D}_{S_0}$ is satisfiable.

3.3.3 Reasoning in Hybrid Temporal Situation Calculus

As discussed in Section 3.2.3, projection is a common computational task that involves determining the truth value of a statement after executing a sequence of actions. Building on the situation calculus, Batusov, De Giacomo, and Soutchanski [2] demonstrated that the concepts of uniform and regressable formulas in the situation calculus [52] can be extended to hybrid temporal situation calculus, and the standard regression operator $\mathcal{R}$, defined for atemporal fluents (see Definition 3.2.3), can be extended to handle temporal fluents. See [2] for details. They then proved a variant of Reiter's regression theorem:

Theorem 3.3.9 (Regression in Stratified Temporal Basic Action Theory)**.** *If $\mathcal{W}$ is a regressable sentence of situation calculus and $\mathcal{D}$ is a stratified temporal basic action theory, then $\mathcal{D} \models \mathcal{W}$ if and only if $D_{S_0} \cup D_{una} \models \mathcal{R}[\mathcal{W}]$.*

Thus to the situation calculus, reasoning about queries in the temporal case can also be reduced to the task of first-order theorem proving.

3.3.4 Example

I use a simple nuclear power plant (NPP) scenario as my running example in this thesis. While a nuclear power plant produces electricity, its core temperature needs to be maintained below a certain threshold by incorporating a cooling system. The cooling system does this by using coolant supplied through pipes attached to it. For simplicity, I assume just one (unnamed) pipe per plant. In this domain, we have the following actions:

1. $rupture(p, t)$, i.e. the pipe in plant p ruptures at time t.

2. $csFailure(p, t)$, i.e. the cooling system of p fails at t.

3. $fixP(p, t)$, i.e. the pipe of plant p is fixed at t.

4. $fixCS(p, t)$, i.e. the cooling system of p is fixed at t.

5. $mRadiation(p, t)$, radiation level of p is measured at t.

The discrete fluents in this domain are:

1. $Ruptured(p, s)$, representing plant p has a ruptured pipe in situation s.

2. $CSFailed(p, s)$, representing the cooling system of p has failed in s.

We also have a temporal functional fluent $coreTemp(p, t, s)$, which stands for the core temperature of power plant p at time t in situation s.

I now give the domain-dependent axioms, starting with the action precondition axioms.

Axiom 3.3.10.

$$a)\ Poss(rupture(p,t),s) \equiv true,$$

$$b)\ Poss(fixP(p,t),s) \equiv Ruptured(p,s),$$

$$c)\ Poss(csFailure(p,t),s) \equiv \neg CSFailed(p,s),$$

$$d)\ Poss(fixCS(p,t),s) \equiv CSFailed(p,s),$$

$$e)\ Poss(mRadiation(p,t),s) \equiv true.$$

These can be read as follows:

(a) a rupture action is always possible;

(b) fixing the pipe of plant p is possible in situation s if and only if the pipe of p is already ruptured in s;

(c) the cooling system failure of plant p action can be executed in situation s if and only if the cooling system of p has not already failed in s;

(d) fixing the cooling system of p is possible in situation s if and only if the cooling system of p has already failed in s; and

(e) the measure radiation of p action has no precondition and is always possible to execute.

We also have the following successor-state axioms for each fluent:

Axiom 3.3.11.

$a)$ $Ruptured(p, do(a, s)) \equiv$

$$\exists t.\ a = rupture(p, t) \vee (Ruptured(p, s) \wedge \neg \exists t.\ a = \mathit{fixP}(p, t)),$$

$b)$ $CSFailed(p, do(a, s)) \equiv$

$$\exists t.\ a = csFailure(p, t) \vee (CSFailed(p, s) \wedge \neg \exists t.\ a = \mathit{fixCS}(p, t)).$$

Thus, Axiom 3.3.11(a) says that the pipe of plant p has ruptured after action a happens in situation s, i.e. in $do(a, s)$, if and only if a is the action of rupturing the pipe of p at some time t, or the pipe of p was already ruptured in s and a does not refer to the action of fixing the pipe of p at some time t. And, according to Axiom 3.3.11(b), the cooling system of plant p has failed after action a happens in situation s, i.e. in $do(a, s)$, if and only if a refers to the failure of cooling system of p action at some time t, or the cooling system of p was already broken in s and a is not the action of fixing the cooling system of p at some time t.

For core temperature, before giving the state evolution axiom, let me list the contexts γ_i, for $i = 1$ to 3:

$$\gamma_1(p) = Ruptured(p) \wedge CSFailed(p),$$

$$\gamma_2(p) = Ruptured(p) \wedge \neg CSFailed(p),$$

$$\gamma_3(p) = \neg Ruptured(p) \wedge CSFailed(p).$$

Now, I give state evolution axiom for temporal fluent $coreTemp(p, t, s)$.

Axiom 3.3.12.

$$coreTemp(p, t, s) = y$$
$$\equiv [(\gamma_1(p, s) \wedge \delta_1(p, t, s)) \vee (\gamma_2(p, s) \wedge \delta_2(p, t, s)) \vee (\gamma_3(p, s) \wedge \delta_3(p, t, s))$$
$$\vee (y = coreTemp(p, start(s), s) \wedge \neg(\gamma_1(p, s) \vee \gamma_2(p, s) \vee \gamma_3(p, s)))].$$

That is, the value of *coreTemp* of p at time t in situation s is dictated by:

1. δ_1 if both p's cooling system has failed and its pipe was ruptured;
2. δ_2 if p's pipe was ruptured but its cooling system is working;
3. δ_3 if p's cooling system has failed but its pipe is intact; and
4. it remains the same as in $start(s)$, otherwise.

Here, δ_i for $i = 1, 2, 3$ is defined as follows:

$$\delta_i(p, t, s) \stackrel{\text{def}}{=} coreTemp(p, t, s) = coreTemp(p, start(s), s) + (t - start(s)) \times \Delta_i,$$

where, Δ_i is the rate of change such that $\Delta_1 = 100, \Delta_2 = 35$, and $\Delta_3 = 55$. The above formula computes $coreTemp(p, t, s)$ by adjusting the initial temperature at $start(s)$ based on the elapsed seconds $t - start(s)$ and specifies a rate of temperature increase, 100, 35, or 55 degrees per second, respectively, depending on the context

γ_i. For simplicity, I use these basic equations, but I could have used more realistic differential equations to model temperature change as well.

We also need the following unique names axioms specifying that the different terms represent different actions (again, these are required by the successor-state axioms):

Axiom 3.3.13.

$$a)\ rupture(p,t) = rupture(p',t') \supset p = p' \wedge t = t',$$

$$b)\ csFailure(p,t) = csFailure(p',t') \supset p = p' \wedge t = t',$$

$$c)\ fixP(p,t) = fixP(p',t') \supset p = p' \wedge t = t',$$

$$d)\ fixCS(p,t) = fixCS(p',t') \supset p = p' \wedge t = t',$$

$$e)\ mRadiation(p,t) = mRadiation(p',t') \supset p = p' \wedge t = t',$$

$$f)\ rupture(p,t) \neq csFailure(p,t),$$

$$g)\ rupture(p,t) \neq fixP(p,t),$$

$$h)\ rupture(p,t) \neq fixCS(p,t),$$

$$i)\ rupture(p,t) \neq mRadiation(p,t),$$

$$j)\ csFailure(p,t) \neq fixP(p,t),$$

$$k)\ csFailure(p,t) \neq fixCS(p,t),$$

$$l)\ csFailure(p,t) \neq mRadiation(p,t),$$

$$m)\ fixP(p,t) \neq fixCS(p,t),$$

$$n)\ fixP(p,t) \neq mRadiation(p,t),$$

$$o)\ fixCS(p,t) \neq mRadiation(p,t).$$

I assume that there is at least one nuclear power plant P_1 in our domain, and add the following initial state axioms for P_1:

Axiom 3.3.14.

$$a)\ \neg Ruptured(P_1, S_0),$$

$$b)\ \neg CSFailed(P_1, S_0),$$

$$c)\ coreTemp(P_1, start(S_0), S_0) = -50.$$

Henceforth, I use $\mathcal{D}_{npp}$ to refer to the above axiomatization .

3.4 Actual Cause in the Situation Calculus

Recall that in Chapter 2, I discussed the limitations of existing approaches to causation, underscoring the necessity for a more robust and expressive framework to effectively model causation in dynamic domains. The study of causation in the situation calculus is proposed as a solution to these challenges due to its ability to model complex and dynamic domains effectively. In this section, I discuss the work done on actual causality in discrete domains within the situation calculus. I use this as a foundation to model actual causality within hybrid domains in subsequent chapters.

3.4.1 Situation Calculus Semantics for Actual Causality

Batusov and Soutchanski [4] proposed a foundational definition of actual causes based on situation calculus action theories. Later, Khan and Soutchanski [32] argued using careful analysis of their proposal that it is possible to get rid of the disagreements between different definitions of actual causality. In particular, they showed that Batusov and Soutchanski's account can be interpreted both as a regularity definition and counterfactually. Batusov and Soutchanski proposed two kinds of causal roles that events may assume: achievement causes, which are events that make the condition of interest, i.e., the effect or its precondition true in whole or in part, and maintenance causes, which are events that prevent other events from making the effect false again. They combined both types of events contributing to the existence of the effect formula as actual causes.

Since all changes in the situation calculus stem from explicit events or actions, the causes of an observed effect φ within a given scenario σ (action history/situation) relative to a domain theory $\mathcal{D}$ must be actions from σ. These elements are combined into the notion of causal setting in the situation calculus, relative to which causes will be computed.

Definition 3.4.1 (Causal Setting)**.** *A causal setting is a tuple $\langle \mathcal{D}, \sigma, \varphi \rangle$, where $\mathcal{D}$ is a basic action theory, σ is a ground situation term such that $\mathcal{D} \models Executable(\sigma)$, and φ is a situation-suppressed situation calculus formula uniform in s such that $\mathcal{D} \models \varphi[\sigma]$.*

This requires the scenario σ to be executable and the effect φ to become true after σ is

executed starting in the initial situation S_0. As mentioned, given a causal setting $\mathcal{C} = \langle \mathcal{D}, \sigma, \varphi \rangle$, causes are actions from the scenario σ. But since σ might include multiple occurrences of the same action, to uniquely identify these actions, the situations where these actions are performed also needed to be identified. Thus in this framework, causes are action-situation pairs.

The idea behind how causes are computed is as follows. Given an effect φ and scenario σ, if some action of the action sequence in σ triggers the formula φ to change its truth value from false to true relative to $\mathcal{D}$, and if there are no actions in σ after it that change the value of φ back to false, then this action is an actual cause of achieving φ in σ. Such causes are referred to as primary causes.

Definition 3.4.2 (Primary Achievement Cause)**.** *If a causal setting $\mathcal{C} = \langle \mathcal{D}, \sigma, \varphi \rangle$ satisfies the following achievement condition via the situation term $do(\alpha, \sigma') \sqsubseteq \sigma$,*

$$\mathcal{D} \models \neg\varphi(\sigma') \wedge \forall s.\ do(\alpha, \sigma') \sqsubseteq s \sqsubseteq \sigma \supset \varphi(s),$$

then α is a primary achievement cause in $\mathcal{C}$.

Note that φ was false in situation σ' and became true immediately in the next situation $do(\alpha, \sigma')$ when α is executed in σ', and then in all subsequent situations, it remained true. This type of cause implies a direct link between the action and the effect. This primary cause might have been non-executable initially in S_0, and thus may require other actions in its history to bring about its preconditions; also, it might bring about the effect only conditionally (or partially), requiring the contribution of

other actions in its history to achieve the condition under which its execution brings about the effect (or to fully achieve the effect, respectively). These actions whose contributions were also necessary to achieve the effect are called secondary or indirect causes.

To compute secondary causes, which are all the necessary actions leading to the achievement of the effect, Batusov and Soutchanski [2] use the achievement condition together with the single-step regression operator $\mathcal{R}$. Formally, $\mathcal{R}[\varphi, \alpha]$ represents the weakest precondition necessary for φ to hold after performing action α in a previous situation σ'. If α is established as an achievement cause of φ in $do(\alpha, \sigma')$, $\mathcal{R}$ generates a formula that is both necessary and sufficient for achieving φ via the execution of α in σ'. Note that $\mathcal{R}[\varphi, \alpha]$ may have its own achievement causes. Additionally, the right-hand side of the action precondition axiom for α, i.e., Π_α captures the conditions required for α to be executed, which may have its own achievement causes as well. In summary, if α is an achievement cause of φ in $do(\alpha, \sigma')$, then $(\mathcal{R}[\varphi, \alpha] \wedge \Pi_\alpha(s))$ expresses the conditions necessary for executing α at σ' and achieving φ via α. Using this, secondary causes are defined as follows:

Definition 3.4.3 (Secondary Causes)**.** *If a causal setting $\mathcal{C} = \langle \mathcal{D}, \sigma, \varphi \rangle$ satisfies the achievement condition via some situation term $do(A(\bar{\mathrm{t}}), \sigma') \sqsubseteq \sigma$ and α is an achievement cause in the causal setting $\langle \sigma', \rho[\varphi, A(\bar{\mathrm{t}})] \wedge \Pi_A(\bar{\mathrm{t}}) \rangle$, then α is an achievement cause in $\mathcal{C}$.*

Recall that the achievement condition requires the effect to be true at the end

of the scenario. Secondary causes do not capture all the events responsible for the effect being true at the end of the scenario. They represent the actions responsible for changing the effect from false to true but not those that prevent it from becoming false again. Consider a scenario in which there exists an action capable of negating φ, i.e., making φ false, after it was achieved by the primary cause. Such actions are referred to as *threat-actions* to the effect. However, φ remains true because other actions, which are called *maintenance actions*, counteract the threat and preserve the effect. Maintenance causes may have their own achievement causes and subsequent maintenance causes. Batusov and Soutchanski [4] define "actual causes" as a set of actions that includes both types of causal roles: achievement causes (primary and secondary causes) and maintenance causes. A detailed discussion of these topics is beyond the scope of this thesis; interested readers are encouraged to refer to [4].

The achievement causal chain proposed in [4] builds on a first-order representation of the dynamic world using the situation calculus and as such can deal with quantified effects. For instance, one can query about the causes of all the blocks being broken, i.e., $\forall x.Broken(x)$. It utilized the regression operator along with achievement and maintenance conditions to uncover a complete sequence of actions leading to the condition of interest.

3.4.2 Embedding the Metatheoretic Account of Actual Causes by Batusov and Soutchanski into the Language of Situation Calculus

Batusov and Soutchanski [4] define actual causes using a syntactic regression operator, while Khan and Lespérance [30] provide a refined semantic definition that accounts for causation from an agent's perspective, incorporating its knowledge. This contrasts with earlier approaches focused solely on objective causation. They showed that besides reasoning about causes and their effects, formalizing knowledge about actual causes can be useful, especially in scenarios where a plan fails and needs to be tailored again. In their framework, an agent can perform a *sensing* action to acquire knowledge about the cause of an effect and re-evaluate the plan. This approach accommodates epistemic causes and effects, allowing agents to analyze the origins of newly acquired knowledge. For instance, an agent may trace the cause of newfound knowledge to specific knowledge-producing actions, such as *inform*. They acknowledge the uncertainty inherent in causal attribution where an agent may have partial knowledge of causes while remaining uncertain about others. This sophisticated modeling of causality enhances our understanding of agent behavior in complex scenarios, especially in multi-agent systems. However, I restrict discussion to objective causality only as my work in hybrid domains does not deal with epistemic effects.

Since causes are computed relative to a given scenario, it is assumed that the scenario is executable, the effect was initially false before any action was executed,

and became true by the end of the scenario, i.e., in causal setting $\langle \mathcal{D}, s, \varphi \rangle$, it must be the case that:

$$\mathcal{D} \models \mathit{Executable}(s) \wedge \neg\varphi[S_0] \wedge \varphi[s]. \tag{3.18}$$

As φ is required to hold by the end of the scenario s, they ignore the cases where φ is not achieved by the actions in s, since if this is the case, the achievement cause truly does not exist.

As discussed, s might include multiple occurrences of the same action. Hence, one also needs to identify the situations where these actions were executed. To deal with this, Khan and Lespérance required that each situation be associated with a time-stamp, which is an integer for their theory. Since in the context of knowledge, there can be different epistemic alternative situations (possible worlds) where an action occurs, using time-stamps provides a common reference/rigid designator for the action occurrence. They 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:

Axiom 3.4.1.

$$\mathit{timeStamp}(S_0) = 0,$$

$$\forall a, s, ts.\ \mathit{timeStamp}(do(a, s)) = ts \equiv \mathit{timeStamp}(s) = ts - 1.$$

With this, causes of a given effect φ in their framework is a non-empty set of action-time-stamp pairs derived from the scenario s.

Khan and Lespérance [30] introduced *dynamic formulae* in the situation calculus, where an effect φ is a situation-suppressed dynamic formula. The notation $\varphi[s]$ denotes that φ is true in situation s, with the situation argument restored in all fluents in φ.

Definition 3.4.4 (Dynamic Formulae)**.** *Let φ range over situation-suppressed formulae, $\vec{x}$ range over object terms, θ_a range over action terms, and $\vec{y}$ range over 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 can be of the following form:

1. $P(\vec{x})$, which is a situation-suppressed fluent.
2. $Poss(\theta_a)$, a formula that says that some action θ_a is possible to execute.
3. $After(\theta_a, \varphi)$, a formula that some dynamic formula holds after some action has occurred.
4. $\neg\varphi$, a formula that does not hold.
5. $\varphi_1 \wedge \varphi_2$, if a conjunction of formulae holds.

Note that φ can have quantification over object and action variables, but must not include quantification over situations or ordering over situations (i.e. $\sqsubset$).

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

Definition 3.4.5.

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

I now give Khan and Lespérance's [30] definition of causes in the situation calculus. 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 direct cause of achieving φ in s. Such causes are referred to as *primary* causes.

Definition 3.4.6 (Primary Cause [30])**.**

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

That is, a executed at time-stamp ts is the *primary cause* of effect φ in situation s if and only if 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. It is different from Batusov and Soutchanski's definition of primary cause (Definition 3.4.2) for the fact that it uniquely identifies actions based on time-stamps attached to each situation.

The following definition can compute both primary and indirect causes, i.e., those that made the primary cause executable or made the effect conditionally or partially true.[2]

[2]In this, we need to quantify over situation-suppressed direct formula. Thus we must encode such formulae as terms and formalize their relationship to the associated situation calculus formulae. This is tedious but can be done essentially along the lines of [14]. We assume that we have such an encoding and use formulae as terms directly.

Definition 3.4.7 (Actual Cause [30])**.**

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

Thus, *Causes* is defined to be the least relation P such that if a executed at time-step ts directly causes φ in scenario s then (a, ts, φ, s) is in P, and if a' executed at ts' is a direct cause of φ in s, the time-stamp of s' is ts', $s' < s$, and $(a, ts, [Poss(a') \wedge After(a', \varphi)], s')$ is in P (i.e. a executed at ts is a direct or indirect cause of $[Poss(a') \wedge After(a', \varphi)]$ in s'), then (a, ts, φ, s) is in P. Here the effect $[Poss(a') \wedge After(a', \varphi)]$ requires a' to be executable and φ to hold after a'.

This inductive definition of causes can be used to handle trickier cases of conditional effects, where some action can bring about an effect only when some condition holds already. Next, I will give an example to illustrate how causes are computed by using Khan and Lespérance's definitions of direct and indirect causes.

3.4.3 Example - Causes in Discrete Case

I illustrate causation in the situation calculus using a variant of my running Example presented in Section 3.3.4. Assume a situation calculus basic action theory $\mathcal{D}_{npp}^{SC}$ for this domain that only includes the atemporal variants of the actions, i.e., *mRadiation*(p), *csFailure*(p), and *fixCS*(p), the fluents *CSFailed*(p, s) and *Ruptured*(p, s), and the associated initial state axioms, action precondition axioms, successor-state axioms, and unique-names axioms; see $\mathcal{D}_{npp}$ in Section 3.3.4. Within this framework, consider causal setting $\mathcal{C} = \langle \mathcal{D}_{npp}^{SC}, \sigma_1, \varphi_1 \rangle$, where

$$\sigma_1 = do([mRadiation(P_1), csFailure(P_1), fixCS(P_1), mRadiation(P_1),$$
$$csFailure(P_1), mRadiation(P_1)], S_0),$$

and $\varphi_1 = CSFailed(P_1, \sigma_1)$, for some powerplant P_1 (this is a constant). In Figure 3.1, I illustrate actions and their effect in scenario σ_1.

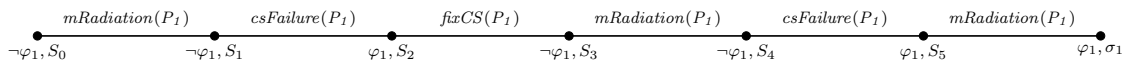

Figure 3.1: Causes in Discrete Case

According to Definition 3.4.6, I can show the following about direct cause in this causal setting.

Proposition 1.

$$\mathcal{D}_{npp}^{SC} \models \mathit{CausesDirectly}(\mathit{csFailure}(P_1), 4, \varphi_1, \sigma_1).$$

That is, $\mathit{csFailure}(P_1)$ executed at time-stamp 4 is a direct cause of φ_1 in scenario σ_1.

Proof sketch. In the following, I use S_1 to represent $do(\mathit{mRadiation}(P_1), S_0)$, S_2 to represent $do(\mathit{csFailure}(P_1), S_1)$, etc. First, note that by Axiom 3.3.10(e), $\mathit{mRadiation}(P_1)$ is executable in S_0. By Axiom 3.3.11(b) (i.e., the successor-state axiom of *CSFailed*), after this action is executed, we have: $\neg\varphi[S_1]$. Also, by Axiom 3.4.1, we have: $\mathit{timeStamp}(S_1) = 1$. Using similar arguments, it can be shown that each of the actions in the scenario σ_1 was executable in their respective situation and φ_1 has the truth value exactly as shown in Figure 3.1. Also, it can be similarly shown that $\mathit{timeStam}(S_4) = 4$.

According to Definition 3.4.6, the last action that made the effect φ_1 from false to true and after which the effect persists is a primary cause. Hence, by the above argument, $\mathit{csFailure}(P_1)$ executed at time-stamp 4 is the direct cause of φ_1. □

Moreover, I can show the following result about (possibly indirect) causes.

Proposition 2.

$$\mathcal{D}_{npp}^{SC} \models \mathit{Causes}(\mathit{csFailure}(P_1), 1, \varphi_1, \sigma_1)$$
$$\wedge\, \mathit{Causes}(\mathit{fixCS}(P_1), 2, \varphi_1, \sigma_1) \wedge \mathit{Causes}(\mathit{csFailure}(P_1), 4, \varphi_1, \sigma_1).$$

Proof sketch. Explaining backward, since by Proposition 1 the second $csFailure(P_1)$ action executed at time-stamp 4 is a direct cause of φ_1 in σ_1, by Definition 3.4.7 it is also a cause of φ_1 in σ_1. Now, it can be shown that $fixCS(P_1)$ executed at timeStamp 2 is a direct cause of $\varphi_2 = Poss(csFailure(P_1) \wedge After(csFailure(P_1), \varphi_1))$ in S_4, and thus by Definition 3.4.7, it is also a cause of φ_1 in σ_1. Finally, it can be shown that $csFailure(P_1)$ executed at timeStamp 1 is a direct cause of $\varphi_3 = Poss(fixCS(P_1) \wedge After(fixCS(P_1), \varphi_2))$ in situation S_2, and thus by Definition 3.4.7, it is also a cause of φ_1 in σ_1. □

3.5 Counterfactual Worlds in Situation Calculus

Counterfactual worlds refer to the worlds that would have been realized had actions/events been different from what actually occurred. These capture hypothetical or contrary-to-fact properties that can be used to explore the consequences of actions in different scenarios. In the context of causation, counterfactuals are often used to justify causal relationships. For example, if event A causes event B (or in our case, the effect φ), a counterfactual statement could be: "If event A had not occurred, then event B (the effect φ) would not have happened (be observed)." Such a statement utilizes the counterfactual scenarios where A did not happen, and helps us reason about the causal impact of specific events, in this case, A. While the Definition 3.4.7 of actual cause appeals to the intuition of causation, others contend that causation should be defined using counterfactuals [27, 26]. In this section, I first give a definition

of what it means for a situation to be counterfactual to another. I then show why a "but for" counterfactual analysis does not work for causes defined in 3.4.7, i.e., study the preemption problem (see Theorem 3.5.2). Note that this is a new contribution to this thesis.

3.5.1 Defining Counterfactual Situations

My notion of counterfactual situations assumes that actions from the given situation are replaced with a different action to produce counterfactual situations. First, I define counterfactual situations that differ from a given situation only by one action.

Definition 3.5.1 (Single-Action Counterfactual Situation)**.**

$$
\begin{aligned}
CF_{one}(s', s) \stackrel{\text{def}}{=} \exists a_1, a_2, s_{sh}.\ & a_1 \neq a_2 \wedge do(a_1, s_{sh}) \sqsubseteq s \wedge do(a_2, s_{sh}) \sqsubseteq s' \\
& \wedge \forall a^*, s^*.\ do(a_1, s_{sh}) \sqsubset do(a^*, s^*) \sqsubseteq s \\
& \quad \supset (\exists s^+.\ timeStamp(s^*) = timeStamp(s^+) \wedge do(a^*, s^+) \sqsubseteq s') \\
& \wedge \forall a^*, s^*.\ do(a_2, s_{sh}) \sqsubset do(a^*, s^*) \sqsubseteq s' \\
& \quad \supset (\exists s^+.\ timeStamp(s^+) = timeStamp(s^*) \wedge do(a^*, s^+) \sqsubseteq s).
\end{aligned}
$$

That is, given a situation s, another situation s' is counterfactual to s and differs from s by just one action if and only if s and s' share a common situation s_{sh} in their history, the actions performed in the history of s and s' in the situation s_{sh} are different, but all other actions in their history (performed before and after s_{sh})

are exactly the same. Here, we use the function $timeStamp(s)$ to ensure that the subsequent actions after the unmatched one are performed in exactly the same order in both histories. Note that, since s_{sh} is a common situation in the history of both s and s', it trivially follows that all actions performed in the history of these scenarios until s_{sh} must be exactly the same.

I will next define a general notion of counterfactual situations. For this, I will use a variant of CF_{one}, $CF_{one}(s', s, \langle a', a, ts\rangle)$, that makes $a = a_1, a' = a_2$, and $ts = timeStamp(s_{sh})$ explicit (these are stored as a triple). Using this, I define counterfactual situations as follows.[3]

Definition 3.5.2 (Counterfactual Situation)**.**

$$CF(s', s, L) \stackrel{\text{def}}{=} \forall P.[\ldots \supset P(s', s, L)],$$

where $\ldots$ stands for

$$\forall s', s, a', a, ts.[CF_{one}(s', s, \langle a', a, ts\rangle) \supset P(s', s, [\langle a', a, ts\rangle])] \wedge$$

$$\forall s'', s, L'.[\exists s', a', a'', ts, L.(CF_{one}(s'', s', \langle a'', a', ts\rangle) \wedge P(s', s, L) \wedge L' = cons(\langle a'', a', ts\rangle, L)$$

$$\wedge \forall a'_1, a_1, ts_1, a'_2, a_2, ts_2.(\langle a'_1, a_1, ts_1\rangle \in L' \wedge \langle a'_2, a_2, ts_2\rangle \in L') \supset ts_1 \neq ts_2)$$

$$\supset P(s'', s, L')],$$

where cons is a standard list function that appends an element to the front of a list. Formally, given an element, in our case $\langle a'', a', ts\rangle$, and a list L, the function $cons(\langle a'', a', ts\rangle, L)$ constructs a new list with $\langle a'', a', ts\rangle$ as the head (first element)

[3]Here, I use the standard list operation *cons* for constructing a new list from an item and a list, so I assume that our theory includes an axiomatization of lists.

and L as the tail (remaining elements).

That is, CF is defined to be the least relation P such that if the counterfactual situation s' can be obtained from s by replacing action a executed at time stamp ts with a', then (s', s, L) is in P, where L is a list that only includes the triple $\langle a', a, ts\rangle$; and if s'' can be obtained from s' by replacing a' executed at time stamp ts with a'', (s', s, L') is in P, L' is the list that can be constructed from triple $\langle a'', a', ts\rangle$ and list L, and each pair of triples in L' has a different time stamp argument, then (s'', s, L') is in P. Note that this does not allow us to obtain a counterfactual situation by replacing actions in the same position twice (as it requires that each triple in the list L' must have a unique time-stamp).

Finally, I also define executable variants that ensure that the counterfactual situation obtained is executable.

Definition 3.5.3 (Executable CF_{one})**.**

$$CFEx_{one}(s', s, \langle a', a, ts\rangle) \stackrel{\text{def}}{=} CF_{one}(s', s, \langle a', a, ts\rangle) \wedge Exectuable(s').$$

Definition 3.5.4 (Executable Counterfactual Situation)**.**

$$CFEx(s', s, L) \stackrel{\text{def}}{=} CF(s', s, L) \wedge Exectuable(s').$$

Before I can give a theorem on how preempted actions can still bring about the effect when a cause is replaced with a *noOp* action which is an action that is always

possible to execute and has no effects, I must state the action precondition axiom for the *noOp* action.

Axiom 3.5.1.

$$Poss(noOp, s) \equiv true.$$

With this, I can show the following result:

Theorem 3.5.2. *[Preempted Contribution]*

$$\mathcal{D} \not\models Causes(a, ts, \phi, s) \supset \neg\exists s'.\ CFEx_{one}(s', s, \langle noOp, a, ts\rangle) \vee \neg\phi(s').$$

Thus, it is not guaranteed that if a executed in timeStamp ts is a cause of φ in scenario s, then either an executable counterfactual situation to s obtained by replacing a at ts by *noOp* does not exist, or the effect φ can no longer be observed in such a counterfactual scenario s'. This indicates that removing the cause will not necessarily make the effect disappear or render the scenario non-executable, as the effect might still follow due to preempted contributors occurring later in the scenario, i.e., actions that would have brought about the effect in the original scenario s had it not for the actual cause a.

Proof of Theorem 3.5.2 *(By counter-example).* This uses our running example's discrete variant in Section 3.4.3 with domain theory $\mathcal{D}^{SC}_{npp}$. Consider a causal setting $\langle \mathcal{D}^{SC}_{npp}, \varphi_2, \sigma_2\rangle$, where the effect φ_2 and the scenario σ_2 are defined as follows (illustrated

in Figure 3.2):

$$\varphi_2 = \mathit{Ruptured}(P_1, \sigma_2),$$

$$\sigma_2 = do([\mathit{rupture}(P_1), \mathit{mRadiation}(P_1), \mathit{fixP}(P_1), \mathit{rupture}(P_1), \mathit{rupture}(P_1)], S_0).$$

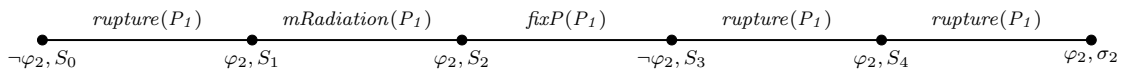

Figure 3.2: Single Action Counterfactual Analysis on Direct Cause

According to Definition 3.4.7, we can show the following result about causes.

Proposition 3.

$$\mathcal{D}^{SC}_{npp} \models \mathit{Causes}(\mathit{rupture}(P_1), 0, \varphi_2, \sigma_2)$$

$$\wedge\, \mathit{Causes}(\mathit{fixP}(P_1), 2, \varphi_2, \sigma_2) \wedge \mathit{Causes}(\mathit{rupture}(P_1), 3, \varphi_2, \sigma_2).$$

Let us replace the primary cause $\mathit{rupture}(P_1)$ executed at time-stamp 3 with $noOp$ action in the scenario σ_2, and call this counterfactual scenario σ_2', which is illustrated in Figure 3.3 below.

$$\sigma_2' = do([\mathit{rupture}(P_1), \mathit{mRadiation}(P_1), \mathit{fixP}(P_1), \mathit{noOP}(), \mathit{rupture}(P_1).$$

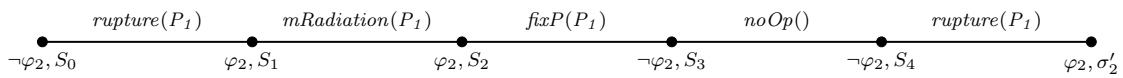

Figure 3.3: Single Action Counterfactual Analysis on Direct Cause

The scenario σ_2' is executable as the *rupture* and *noOp* actions are possible to execute according to Axioms 3.3.10(a) and 3.5.1. It is also easy to verify using Axiom 3.3.11(a) that the effect still holds at the end of σ_2'. Thus

$$\mathcal{D}_{npp}^{SC} \models CFEx_{one}(\sigma_2', \sigma_2, \langle noOp, rupture(P_1), 3\rangle) \wedge \varphi_2[\sigma_2'].$$

□

In this section, I defined the notions of single-action and multiple-action counterfactual situations relative to a given scenario. Through an illustrative example, I highlighted the issue of preemption when defining causes based on counterfactual scenarios and "but for" analysis. Note that I could have removed all preempted causes from the scenario before checking if the effect still follows, and thus a variant of "but for" cause analysis might still work. I will return to this issue when discussing the temporal case in Chapter 5, which is the main focus of this thesis.

3.6 Conclusion

In this chapter, I reviewed previous work on formalizing dynamic domains, focusing on the situation calculus as a tool for reasoning about actions and their effects. I discussed the hybrid temporal situation calculus, an extension that accommodates both discrete and continuous change, which will be foundational for defining the notion of primary cause in hybrid dynamic domains in Chapter 4. Additionally, I

examined the formalization of achievement causes in discrete domains within the situation calculus. After discussing the proposal by Batusov and Soutchanski [4], and the embedding by Khan and Lespérance [30], I proposed a definition of counterfactual worlds, which will be extended to hybrid domains in Chapter 5.

Chapter 4

A Foundational Definition of Primary Achievement Cause in Hybrid Dynamic Domains

4.1 Introduction

Motivated by the hybrid nature of real-world actions and their effects, in this chapter, I study causation in the hybrid temporal situation calculus. As discussed in the previous chapter, while others have explored causation in the situation calculus, these approaches are limited to discrete domains. In contrast, in hybrid domains, changes can be both discrete and continuous.

I start by defining a notion of a proper hybrid causal setting. Following this, I examine causal settings where the effect is a *primitive* atemporal fluent, as well as those where it refers to a *primitive* temporal fluent, and thus my main proposal

here focuses on primitive effects exclusively. My proposal is based on Khan and Lespérance's definition of primary cause [30]. In addition, I also (informally) hint at how my definition of primary cause can be extended to cover effects consisting of conjunctions and disjunctions of two or more primitive temporal fluents. I use a simple nuclear power plant example to illustrate my definition of primary cause and its conjunctive and disjunctive variants.

In the subsequent section, I also explore some basic properties of my definition, including one that identifies the conditions under which the primary cause persists. To support our discussion, I use an example to illustrate these properties.

4.2 Setting the Stage

Causal Setting

I now define a notion of causal setting in the hybrid temporal situation calculus.

Definition 4.2.1 (Hybrid Temporal Achievement Causal Setting)**.** *A hybrid temporal achievement causal setting is a tuple* $\langle \mathcal{D}, \sigma, \varphi \rangle$*, where* $\mathcal{D}$ *is a hybrid temporal situation calculus basic action theory,* $\sigma \neq S_0$ *is a ground situation term of the form* $do([\alpha_1, \ldots, \alpha_n], S_0)$ *with non-empty sequence of ground action functions* $\alpha_1, \ldots, \alpha_n$*, and* φ *is a situation-suppressed (possibly temporal, and in that case, time-suppressed) situation calculus formula that is uniform in* s *such that:*

$$\mathcal{D} \models \mathit{Executable}(\sigma) \wedge \neg\varphi[\mathit{start}(S_0), S_0] \wedge \neg\varphi[\mathit{time}(\alpha_1), S_0] \wedge \varphi[\mathit{start}(\sigma), \sigma].$$

Thus if $\langle \mathcal{D}, \sigma, \varphi \rangle$ is a hybrid temporal achievement causal setting (hybrid setting, henceforth), then the scenario σ must be executable, the effect φ must be false initially and must remain false till the end of initial situation S_0, and φ is assumed to be true at the beginning of the final situation, i.e., σ. Note that in hybrid temporal situation calculus, it is possible for some context to be true initially in S_0, and in that case, the effect can be achieved between the start and end times of the initial situation without requiring contribution from any action in the scenario. To rule out such cases, my definition requires the effect φ to be false at the beginning and at the end of the initial situation S_0. As we will see later, this restriction is not strong enough, and in some contexts it is indeed still possible for an effect to become true despite having no contributing actions, e.g., when the context that brought about the effect was true initially.

To ensure that the effect is actually achieved within the scenario, I also require the effect to be true when observed at the beginning of σ. In a hybrid domain, in general, one can query the causes of an observed effect at any time-point within a situation σ, i.e. at any time-point in between the start-time and the end-time of σ, inclusive. To simplify, I assume that the query is posed relative to the starting time of σ (as enforced by Definition 4.2.1). If this is not the case, one can always add a subsequent dummy action, $noOp$ (which has no effect and is always possible to execute), and query with respect to the updated scenario $do(noOp, \sigma)$, and hybrid setting $\langle \mathcal{D}, do(noOp, \sigma), \varphi \rangle$.

In my framework, φ is a situation- and time-suppressed hybrid temporal situation

calculus formula. The exact nature of φ is irrelevant for the task at hand as in this thesis I only deal with primitive atemporal fluents and conditions on the values of primitive temporal fluents (e.g. $coreTemp(P) > 1000$) and restricted extensions of these as effects. From now on, I will write $\varphi[t, s]$ to denote the formula obtained from φ by restoring the appropriate situation and time arguments into the only fluent in φ, and thus, for example, $coreTemp(P_1)[5, S_0]$ stands for $coreTemp(P_1, 5, S_0)$. As discussed above, a hybrid setting does not necessarily guarantee that the causes of the associated (temporal) effect can always be computed as these might still be implicit in the initial situation, e.g., when the context that brought about the effect was true initially and remained true until the achievement of the effect; we will return to this issue later in Theorem 4.5.4.

One last point: when the effect is atemporal, the time arguments in Definition 4.3.1 above are simply ignored and the definition of hybrid setting resembles that of a causal setting in the situation calculus.

Atemporal Primitive Fluents as Effects.

I next consider defining primary achievement cause relative to a hybrid setting, where the effect is a primitive atemporal fluent. Since we already have a notion of time associated with every situation in the hybrid temporal situation calculus (see Axiom 3.3.2 in Chapter 3), it seems natural to adopt this instead of Khan and Lespérance's [30] time-stamp. However, one issue with this approach is that, because actions in hybrid temporal situation calculus are instantaneous, multiple actions can

occur at the same time-point. Consequently, the execution time of an action a, represented using $time(a)$, cannot be used to uniquely identify the action within a a given scenario/trace. Therefore, I adopt Khan and Lespérance's time stamps and as a result, for defining causes of atemporal primitive fluents, their Definition 3.4.6 above. I require that φ in this case be some suitable subclass of hybrid temporal situation calculus formulae.

Temporal Primitive Fluents as Effects.

For discrete effects, the primary cause, which is an action a executed at time-point t, brings about the effect discretely and immediately after the execution of a at t. For the temporal case, however, the effect might be only realized after a while, and one or more irrelevant actions might be executed in between. For example, if one changes the temperature on a thermostat, the desired room temperature will likely be achieved after some time has passed, but in between there can be other irrelevant actions that might be executed, those that have no impact on the value of the room temperature. Thus, while defining the primary achievement cause, in addition to actions causing the change in a temporal fluent's value, we need to identify the situation where the effect was actually achieved within the scenario.

4.3 Primary Achievement Cause: The Primitive Case

4.3.1 Intuition and Definition

Let me start with the intuition behind my definition. Recall that in hybrid temporal situation calculus, the values of temporal fluents can change only when certain relevant contexts are enabled. Contexts for a temporal fluent, which are (mutually exclusive) discrete fluents, on the other hand are enabled or disabled due to the execution of actions. Thus, when determining the primary cause of some temporal fluent having a certain value, we first need to identify the last context γ that was enabled before the fluent acquired this value, i.e., the context γ which was true in the achievement situation s_φ of the effect φ, and then figure out the action a that caused/enabled this context in s_φ. Since contexts are mutually exclusive (no two contexts can be true at the same time), γ must have been the only enabled context in the achievement situation s_φ, which ensures that the action a is unique. Additionally, a must have been the last action that enabled γ, and whose contribution brought about the temporal effect under consideration.[1]

In the following, I give the definition of primary cause relative to a hybrid causal setting $\langle \mathcal{D}, \sigma, \varphi \rangle$. In this, the effect φ is a constraint on the values of a situation- and time-suppressed primitive temporal fluent $f(\vec{x})$. Also, γ_i^f refers to the contexts $\delta_i(\vec{x}, y, t, s)$, indexed by i, that are associated with the temporal fluent f (see state

[1]There can certainly be other secondary/indirect causes, but I am only concerned with primary causes in this thesis.

evolution axioms defined in Section 3.3.2 above).

Definition 4.3.1 (Primary Achievement Cause (Primitive Temporal Case))**.**

$$\begin{aligned} &\mathit{CausesDirectly}^{\mathit{prim}}_{\mathit{temp}}(a, ts, \varphi, s) \stackrel{\text{def}}{=} \\ &\qquad \exists s_{\varphi}.\ \mathit{AchvSit}(s_{\varphi}, \varphi, s) \wedge \exists i.\ \mathit{CausesDirectly}(a, ts, \gamma_i^f, s_{\varphi}). \end{aligned}$$

That is, an action a executed at time-stamp ts directly causes the effect φ in scenario s if and only if the achievement situation of φ in s is s_{φ}, and a executed in some earlier situation with time-stamp ts directly caused the active context γ_i^f for the temporal fluent f mentioned in φ in scenario s_{φ}. Here, $\mathit{CausesDirectly}(a, ts, \gamma_i^f, s_{\varphi})$ is the same as defined by Khan and Lespérance (see Definition 3.4.6). Note that, $\mathit{CausesDirectly}(a, ts, \gamma_i^f, s_{\varphi})$ implies that the context γ_i^f holds in s_{φ}, i.e., $\gamma_i^f[s_{\varphi}]$, and thus it is indeed the (unique) context that was active in s_{φ}. Since contexts are mutually exclusive, we do not need to check whether a subsequent action executed after a made another context true before the achievement of the effect in situation s_{φ}.

Achievement Situation

I now formally define the achievement situation[2]. First, let me define a function that provides the end time of a situation s' within a given scenario s. Since it is not directly possible to talk about the end time of a situation in the hybrid temporal

[2]A variant of this definition appeared in our award-winning Canadain AI 2024 paper [42], which I later found to be problematic/incomplete; the following definition fixes the issue.

situation calculus (as the end time does not really exist when the scenario is not known), I will use the start time of the next action within the scenario to denote this.

Definition 4.3.2 (End Time of a Situation within a Context)**.**

$$end(s', s) \stackrel{\text{def}}{=} \begin{cases} start(s') & \text{if } s' = s \\ time(a) & \text{if } \exists a.\ do(a, s') \leq s \end{cases}$$

That is, the end time of a situation s' in scenario s is the starting time of s' if s' is the last situation in scenario s, or the time of the execution of the next action a in s' within scenario s, i.e., $time(a)$ such that $do(a, s') \leq s$. Note that, since my definition of hybrid setting guarantees that causes are computed relative to the starting time of the scenario, taking the starting time of s' as the end time of it is reasonable when $s' = s$. Also, while in what follows, I mention that the end time of the situation comes "right before" the execution time of the next action on the trace, note that in reality, these two times are the same.[3]

To define the achievement situation, observe that the effect φ must be true at the end of the achievement situation and must remain true in all subsequent situations and times. But since there can be multiple situations in between the achievement situation and the final situation in the scenario, we must also ensure that the achievement situation is the earliest such situation to uniquely identify it. The following definition captures this intuition.

[3]This is not to say that we allow concurrent actions.

Definition 4.3.3 (Achievement Situation)**.**

$$
\begin{aligned}
&\mathit{AchvSit}(s_\varphi, \varphi, s) \stackrel{\text{def}}{=} \\
&\quad \varphi[\mathit{end}(s_\varphi, s), s_\varphi] \wedge \forall s', t.\ s_\varphi < s' \leq s \wedge \mathit{start}(s') \leq t \leq \mathit{end}(s', s) \supset \varphi[t, s'] \\
&\quad \wedge (\neg \exists s''.\ s'' < s_\varphi \wedge \varphi[\mathit{end}(s'', s), s''] \wedge \forall s', t.\ s'' < s' \leq s \wedge \mathit{start}(s') \leq t \leq \mathit{end}(s', s) \\
&\qquad \supset \varphi[t, s']).
\end{aligned}
$$

That is, s_φ is the achievement situation of the effect φ in scenario s if and only if φ holds at the end of the situation s_φ, and φ continues to hold in all subsequent situations s' and time points t between $\mathit{start}(s')$ and $\mathit{end}(s', s)$. Additionally, there must not exist another preceding situation s'' before s_φ that satisfies these conditions. This ensures that s_φ is the earliest situation in which φ is achieved and maintained till the start of the scenario s.

4.4 Handling Compound Effects

So far, I have only dealt with "primitive" temporal effects, i.e., effects with conditions on the value of a single temporal fluent. In this section, I present preliminary work aimed at identifying the primary cause for disjunction and conjunction of two primitive temporal fluents. Later in Section 4.6, I will provide examples to illustrate these definitions.

Conjunction of Primitive Temporal Fluents

First, consider an effect of the form of $\varphi_a \wedge \varphi_b$, where φ_a and φ_b are primitive temporal effects. Suppose there exists an action a executed at timestamp t_a that directly causes φ_a, and another action b executed at timestamp t_b that directly causes φ_b. I call action a a direct cause of $(\varphi_a \wedge \varphi_b)$ if and only if a is executed after b, i.e., $t_b \leq t_a$. Recall that the primary cause is the most recent action responsible for the achievement of the effect. Since achieving $(\varphi_a \wedge \varphi_b)$ requires both φ_a and φ_b to be true, the action with the later timestamp—in this case, action a—must be considered as the direct cause. Note that when $t_a = t_b$, a and b must refer to the same action. This yields the following definition.

Definition 4.4.1 (Primary Cause of Conjunction of Temporal Effects)**.**

$$\begin{aligned} \mathit{CausesDirectly}_{temp}(a, t_a, \varphi_a \wedge \varphi_b, s) \stackrel{\text{def}}{=} \exists b, t_b.\ & \mathit{CausesDirectly}^{prim}_{temp}(a, t_a, \varphi_a, s) \\ & \wedge \mathit{CausesDirectly}^{prim}_{temp}(b, t_b, \varphi_b, s) \wedge t_b \leq t_a. \end{aligned}$$

Disjunction of Primitive Temporal Fluents

In the case of a disjunction where the effect is of the form $\varphi_a \vee \varphi_b$, where φ_a and φ_b are primitive temporal fluents, it is possible that the overall effect $(\varphi_a \vee \varphi_b)$ can be brought about/caused by the achievement of φ_a or φ_b alone. Similarly, it is also possible for the overall effect to be brought about by the simultaneous achievement of φ_a and φ_b. Consequently, taking the most recent action, say a, that achieved a

context γ_a^f related to the fluent in φ_a to be the primary cause is problematic as it is possible that the overall effect was actually caused by the achievement of φ_b (and not that of φ_a). For instance, consider a scenario where action b is the latest action that enabled a context for φ_b, but φ_b never became true, and $(\varphi_a \vee \varphi_b)$ was ultimately achieved because φ_a became true as a result of an earlier action a (see Example 4.6.4). Here, although a occurred before b, a is indeed the cause for the effect. This observation suggests the following definition.

Definition 4.4.2 (Primary Cause of Disjunction of Temporal Effects)**.**

$$
\begin{aligned}
&\mathit{CausesDirectly}_{temp}(a, t_a, \varphi_a \vee \varphi_b, s) \stackrel{\text{def}}{=} \\
&\quad \exists s_a.\ \mathit{CausesDirectly}^{prim}_{temp}(a, t_a, \varphi_a, s) \wedge \mathit{AchvSit}(s_a, \varphi_a, s) \\
&\quad \wedge [\exists s_b, b, t_b.\ \mathit{CausesDirectly}^{prim}_{temp}(b, t_b, \varphi_b, s) \wedge \mathit{AchvSit}(s_b, \varphi_b, s) \\
&\quad\quad \supset (s_a < s_b \vee (s_a = s_b \wedge t_b < t_a))].
\end{aligned}
$$

That is, an action a is the primary cause of $\varphi_a \vee \varphi_b$ if and only if a executed at time t_a is the direct cause of φ_a, s_a is the achievement situation of φ_a, and if φ_b was also achieved and thus has the achievement situation s_b, then either s_a is earlier than s_b (i.e., $s_a < s_b$), or φ_b has the same achievement situation as φ_a (i.e., $s_a = s_b$) and a was executed after b (i.e., $t_b < t_a$).

The ideas presented above are preliminary and I plan to study these definitions and look into their consequences in the future.

4.5 Properties

I now prove some intuitive properties of my formalization of primary cause. First, given a causal setting, the direct causes of discrete effects are unique.

Lemma 4.5.1 (Uniqueness of Direct Cause)**.** *Given a causal setting* $\langle \mathcal{D}, \varphi, s \rangle$*, it follows that:*

$$\mathcal{D} \models \forall a, a', ts, ts'.\ \mathit{CausesDirectly}(a, ts, \varphi, s) \wedge \mathit{CausesDirectly}(a', ts', \varphi, s)$$
$$\supset a = a' \wedge ts = ts'.$$

Proof. Follows directly from Definition 3.4.6. □

Next, the achievement situation for a given hybrid setting is unique.

Lemma 4.5.2 (Uniqueness of Achievement Situation)**.** *Given a hybrid setting* $\langle \mathcal{D}, \varphi, s \rangle$*, we have:*

$$\mathcal{D} \models \mathit{AchvSit}(s_\varphi, \varphi, s) \wedge \mathit{AchvSit}(s'_\varphi, \varphi, s) \supset s_\varphi = s'_\varphi.$$

Proof. Follows trivially from Definition 4.3.3. □

Moreover, the direct cause of primitive temporal fluents is unique.

Theorem 4.5.3 (Uniqueness of Primary Cause of Temporal Effects)**.** *Given a hybrid*

setting $\langle \mathcal{D}, \varphi, s \rangle$*, we have:*

$$\mathcal{D} \models \mathit{CausesDirectly}^{prim}_{temp}(a_1, ts_1, \varphi, \sigma) \wedge \mathit{CausesDirectly}^{prim}_{temp}(a_2, ts_2, \varphi, \sigma)$$
$$\supset a_1 = a_2 \wedge ts_1 = ts_2.$$

This says that if action a_1 executed at time-stamp ts_1 is a direct cause of effect φ in scenario σ, and action a_2 executed at time-stamp ts_2 is also a direct cause of φ in σ, then a_1 and a_2 must refer to the same action, and ts_1 must be equal to ts_2.

Proof. Follows from the definition of primary achievement cause (Definition 4.3.1), the uniqueness of the achievement situation s_φ (Lemma 4.5.2), the mutual exclusivity of contexts (Axiom 3.3.4), and the uniqueness of direct causes (Lemma 4.5.1). □

Next, as mentioned above, the primary causes of primitive temporal fluents might not exist (as it may be implicit in the initial situation S_0). This holds even if the causal setting under consideration is a proper hybrid setting (as specified by Definition 4.2.1). To see this, assume that a context was already enabled in the initial situation, capturing an ongoing change; then without requiring the contribution of any relevant actions in the scenario, this context can independently achieve the effect if it is not disabled before the achievement of the effect and if this achievement is maintained till the end of the trace. I illustrate this in Example 4.6.5.

Theorem 4.5.4 (Implicit Primary Cause)**.** *Assume that* φ *is a constraint on the*

value of a primitive temporal fluent f. Then we have:

$$\mathcal{D} \models (\mathit{ProperHTSCAchvCausalSetting}(\varphi, \sigma)$$

$$\wedge\, \exists s_{\varphi}.\ \mathit{AchvSit}(s_{\varphi}, \varphi, \sigma) \wedge \exists i.\ \gamma_i^f[s_{\varphi}] \wedge (\forall s'.\ S_0 \leq s' \leq s_{\varphi} \supset \gamma_i^f[s'])$$

$$\supset \neg\exists a, ts.\ \mathit{CausesDirectly}_{temp}^{prim}(a, ts, \varphi, \sigma)),$$

where,

$$\mathit{ProperHTSCAchvCausalSetting}(\varphi, \sigma) \stackrel{\text{def}}{=} \mathit{Executable}(\sigma)$$

$$\wedge\exists a_0.\ do(a_0, S_0) \leq \sigma \wedge \neg\varphi[start(S_0), S_0] \wedge \neg\varphi[time(a_0), S_0] \wedge \varphi[start(\sigma), \sigma].$$

This states that in a given proper hybrid causal setting $\langle \mathcal{D}, \sigma, \varphi \rangle$, if there exists a context γ_i^f such that γ_i^f was true in the initial situation S_0 and remained true until the effect φ was achieved in the achievement situation s_{φ} in σ, then the primary cause of temporal effect φ in σ simply does not exist.

Proof (by contradiction). Fix $\varphi_1, \sigma_1, s_{\varphi_1}, \gamma_{i_1}^f$ and assume that:

$$\mathit{AchvSit}(s_{\varphi_1}, \varphi_1, \sigma_1) \wedge \gamma_{i_1}^f[s_{\varphi_1}], \tag{4.19}$$

$$\forall s'.S_0 \leq s' \leq s_{\varphi_1} \supset \gamma_{i_1}^f[s']. \tag{4.20}$$

Fix a_1 and ts_1 and also assume that:

$$\mathit{CausesDirectly}_{temp}^{prim}(a_1, ts_1, \varphi_1, \sigma_1). \tag{4.21}$$

Now, note that by 4.19 and Lemma 4.5.2, the achievement situation s_{φ_1} is unique. Thus by Axiom 3.3.4 (which guarantees that the contexts are mutually exclusive) and the definition of primary achievement cause in primitive temporal case (Definition 4.3.1), we have:

$$\mathit{CausesDirectly}(a_1, ts_1, \gamma^f_{i_1}, s_{\varphi_1}). \tag{4.22}$$

But this is contradictory to 4.20 and the definition of direct cause (Definition 3.4.6).

□

Finally, I study the conditions under which primary achievement causes persist when the scenario changes. To this end, I first show a result about the persistence of achievement situations.

Lemma 4.5.5. *Given a hybrid setting $\langle \mathcal{D}, \varphi, s \rangle$, we have:*

$$\begin{aligned}\mathcal{D} \models\ & \mathit{AchvSit}(s_\varphi, \varphi, s) \wedge s < s^* \\ & \wedge (\forall s', t.\ s \leq s' \leq s^* \wedge \mathit{start}(s') \leq t \leq \mathit{end}(s', s^*) \supset \varphi[t, s']) \\ & \supset \mathit{AchvSit}(s_\varphi, \varphi, s^*).\end{aligned}$$

Proof. Follows from antecedent and Definition 4.3.3. □

Note that by Definition 4.3.3, the achievement situation is the earliest situation where the effect becomes true and remains true thereafter. It is not difficult to see that if the

scenario is extended and the effect remains true throughout the extended scenario, the achievement situation remains the same in the extended scenario. Using this, I can show the following result.

Theorem 4.5.6 (Persistence)**.** *Given a hybrid setting $\langle \mathcal{D}, \varphi, s \rangle$, we have:*

$$\begin{aligned} \mathcal{D} \models\ & \mathit{CausesDirectly}^{prim}_{temp}(a, ts, \varphi, s) \\ & \wedge (\forall s', t'.\ s \leq s' \leq s^* \wedge \mathit{start}(s') \leq t' \leq \mathit{end}(s', s^*) \supset \varphi[t', s']) \\ & \supset \mathit{CausesDirectly}^{prim}_{temp}(a, ts, \varphi, s^*). \end{aligned}$$

That is, if an action a executed at time-stamp ts is the primary cause of a temporal effect φ in scenario s, then a remains the primary cause of φ in all subsequent situations/scenarios s^* as long as φ does not change after it is achieved in s. Note that this holds even if the context changes and the value of the associated fluent f in φ varies, provided that φ itself remains unchanged.

Proof. By Lemma 4.5.5 and the antecedent, the achievement situation s_φ in s and s^* remains the same. The property thus follows from this, the definition of Primary Achievement Cause in Primitive Temporal Case (Definition 4.3.1), the uniqueness of achievement situations (Lemms 4.5.2), the mutual exclusivity of contexts (Axiom 3.3.4), and the uniqueness of direct cause (Lemma 4.5.1), which together ensures that the unique context associated with the unique achievement situation s_φ in s and s^* is unique. □

4.6 Examples

To illustrate the proposed definition of primary cause for primitive temporal case and its conjunctive and disjunctive variants, in this Section, I will present three examples using domain theory D_{npp} and D_{npp2}. I conclude with an example where the primary cause is implicit in a proper hybrid causal setting.

4.6.1 Example: The Primitive Temporal Case

First, I give an example of the primitive temporal case, where the effect is simply a constraint on the values of a single primitive temporal fluent. Consider the causal setting $\langle \mathcal{D}_{npp}, \varphi_3, \sigma_3 \rangle$, where the effect φ_3 and the scenario σ_3 are defined as follows:

$$\varphi_3 \stackrel{\text{def}}{=} \mathit{coreTemp}(P_1) \geq 1000,$$

$$\sigma_3 = \mathit{do}([\mathit{rupture}(P_1, 5), \mathit{csFailure}(P_1, 15), \mathit{mRadiation}(P_1, 20), \mathit{fixP}(P_1, 26)], S_0).$$

Recall that, the last argument of each action represents the execution time of that action. This scenario is depicted in Figure 4.1, which also shows the temperature at the beginning and at the end of each situation for clarity.

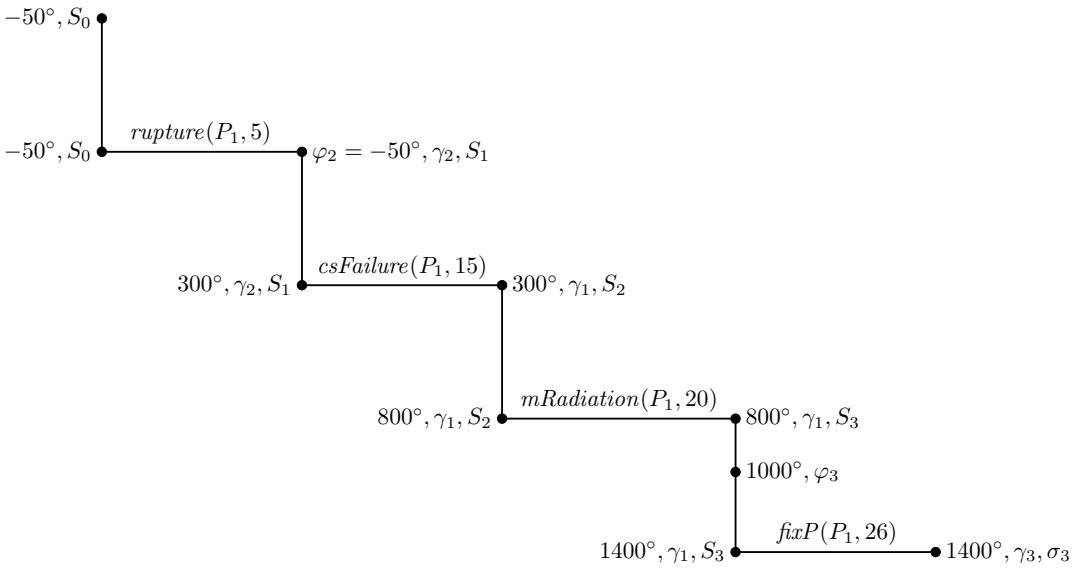

Figure 4.1: Primary Cause in Hybrid Domains: Primitive Case

In this domain, it can be shown that:

Proposition 4.

$$\mathcal{D}_{npp} \models \mathit{CausesDirectly}(\mathit{csFailure}(P_1, 15), 1, \gamma_1, S_3).$$

Proposition 5.

$$\mathcal{D}_{npp} \models \mathit{CausesDirectly}^{prim}_{temp}(\mathit{csFailure}(P_1, 15), 1, \varphi_3, \sigma_3).$$

The given causal setting is a proper achievement causal setting because all actions in the scenario are executable in the order of their execution. Initially, the core temperature is $-50°$, and by the end of the scenario, it reaches $1400°$, thus satisfying the achievement condition.

Proof. By Axiom 3.3.14(a) and (b), since no contexts among $\gamma_1, \gamma_2, \gamma_3$ were active in S_0, by Axiom 3.3.12 and 3.3.14(c) the temperature of P_1 remains $-50°$ in S_0 at time 5.

In situation S_0, the $rupture(P_1, 5)$ action is executed at time 5. According to Axiom 3.3.10(a), this action was executable in S_0. Moreover, as specified in Axiom 3.3.11(a) its execution makes $Ruptured(P_1, S_1)$ true, where $S_1 = do(rupture(P_1, 5), S_0)$. Moreover, by Axiom 3.3.14(b) and 3.3.11(b), we have $\neg CSFailed(do(rupture(P_1, 5), S_0))$. Thus in this situation, the context γ_2 is true. By Axiom 3.3.12, this initiates the increase in *coreTemp* as per δ_2, i.e., the temperature changes $35°$ per second. The core temperature thus reaches $300°$ at time 15.

Next, the $csFailure(P_1, 15)$ action is executed. As shown above, $\neg CSFailed(P_1, S_1)$ holds. Consequently, $csFailure(P_1, 15)$ was possible to execute as per Axiom 3.3.10(c). After the action $csFailure(P_1, 15)$ is executed at time 15, as specified by Axiom 3.3.11(b), $CSFailed(P_1, S_2)$ becomes true, where $S_2 = do(csFailure(P_1, 15), S_1)$. In addition, $Ruptured(P_1, S_2)$ also holds, which activates the γ_1 context. By Axiom 3.3.12, this action continues to change the *coreTemp*, but with a different rate of change (i.e., $\Delta_1 = 100$). The core temperature thus reaches $800°$ at time 20.

After the execution of the $mRadiation(P_1, 20)$ action which was also possible to execute according to Axiom 3.3.10(e), $Ruptured(P_1, S_3)$ and $CSFailed(P_1, S_3)$ continue to hold according to Axiom 3.3.11(a) and (b) and thus the context (γ_1) remains the same and so does the change rate. The effect φ_3 is achieved within situation S_3, where $S_3 = do(mRadiation(P_1, 20), S_2)$.

coreTemp continues to change with Δ_1 rate and reaches 1400 as γ_1 is still active. After $rupture(P_1, 5)$ is executed in situation S_0, we have $Ruptured(P_1, S_1)$, and after the execution of $csFailure(P_1, 15)$ and $mRadiation(P_1, 20)$, we have $Ruptured(P_1, S_3)$, as specified in Axiom 3.3.11(b). Finally, the $fixP(P_1, 26)$ action is also possible to execute in situation S_3 as per Axiom 3.3.10(c), which requires $Ruptured(P_1, S_3)$ for its execution. As per Axiom 3.3.12 and 3.3.11(a) the context γ_3 becomes active in σ_3.

The execution of all action in σ_3 resembles that shown in Figure 4.1. Thus, by definition of *AchvSit*, it follows that:

$$\mathcal{D}_{npp} \models AchvSit(S_3, \varphi_3, \sigma_3) \wedge \gamma_1[S_3]. \tag{4.23}$$

According to the definition of the primary cause for primitive temporal case (Definition 4.3.1), the direct cause (see Definition 3.4.6) of context γ_1 is the primary cause of the effect φ_3. It can be shown that $csFailure(P_1, 15)$ is this cause. □

Note that, although $mRadiation(P_1, 20)$ is the latest action before achieving φ_3 in situation S_3, it is irrelevant because it did not enable any context and thus did not contribute to the change in *coreTemp*. My definition correctly identifies $mRadiation(P_1, 20)$ as an irrelevant action. Also, even though $rupture(P_1, 5)$ contributed to the change, it is not a primary cause of φ_3 because it is not the primary cause of the enabled context γ_1 in achievement situation S_3.

4.6.2 Examples: Compound Cases

Before presenting examples of conjunctive and disjunctive cases, I will extend the domain theory of our running example. I will add two more actions: $fuelMisH(p,t)$ (fuel mishandling of plant p at time t) and $fuelCl(p,t)$ (fuel cleanup of p at t). Additionally, I will introduce two more fluents: a discrete fluent $FuelMH(p,s)$ (indicating that fuel of p is mishandled in s) and a temporal fluent $radLevel(p,t,s)$ (denoting the radiation level of p at time t in situation s).

I add two action precondition axioms to our theory:

Axiom 4.6.1.

$$a)\ Poss(fuelMisH(p,t),s) \equiv true,$$

$$b)\ Poss(fuelCl(p,t),s) \equiv FuelMH(p,s).$$

That is, $fuelMisH(p,t)$ can always be executed, while $fuelCl(p,t)$ can only be executed in situation s if fuel mishandled is true in that situation.

Next, I add a successor-state axiom for the $FuelMH(p,s)$ fluent:

Axiom 4.6.2.

$$FuelMH(p,do(a,s)) \equiv \exists t.\ a = fuelMisH(p,t) \vee (FuelMH(p,s) \wedge a \neq fuelCl(p,t)).$$

That is, the fuel of plant p is mishandled in situation $do(a,s)$ if a refers to a

fuelMisH(p, t) action for p at some time t, or if the fuel of p was already mishandled in s and a does not refer to a cleanup of p action.

And for temporal fluent *radLevel*(p, t, s), I add the following state evolution axiom to describe how it changes over time under the only relevant context *FuelMH*(p).

Axiom 4.6.3.

$$\begin{aligned} \mathit{radLevel}(p, t, s) = y \equiv [&(\mathit{FuelMH}(p, s) \wedge \delta_4(p, t, s)) \\ &\vee (y = \mathit{radLevel}(p, \mathit{start}(s), s) \wedge \neg \mathit{FuelMH}(p, s)))], \end{aligned}$$

where,

$$\delta_4(p, t, s) \stackrel{\text{def}}{=} \mathit{radLevel}(p, t, s) = \mathit{radLevel}(p, \mathit{start}(s), s) + (t - \mathit{start}(s)) \times 2.$$

That is, if the context *FuelMH*(p, s) is true, then the radiation level changes at the rate of δ_4 (an increase of two units each second); otherwise, it remains the same as at the beginning of the situation.

Finally, I add the following two initial state axioms:

Axiom 4.6.4.

$$a)\ \neg \mathit{FuelMH}(P_1, S_0),$$

$$b)\ \mathit{radLevel}(P_1, \mathit{start}(S_0), S_0) = 0.$$

I call this extended domain theory $\mathcal{D}_{npp2}$, defined as follows:

$$\mathcal{D}_{npp2} = \mathcal{D}_{npp} \cup \{\text{Axiom4.6.1 — 4.6.4}\}. \tag{4.24}$$

4.6.3 Example: Conjunctive Case

Consider the causal setting $\langle \mathcal{D}_{npp2}, \varphi_4, \sigma_4 \rangle$, where:

$\varphi_4 = (coreTemp(P_1) \geq 1000) \wedge (radLevel(P_1) \geq 30)$,

$\sigma_4 = do([rupture(P_1, 10), fuelMisH(P_1, 11), csFailure(P_1, 20), mRadiation(P_1, 25),$

$fixP(P_1, 31)], S_0)$.

This is depicted in Figure 4.2, which also shows the core temperature and radiation levels at the beginning and end of each situation.

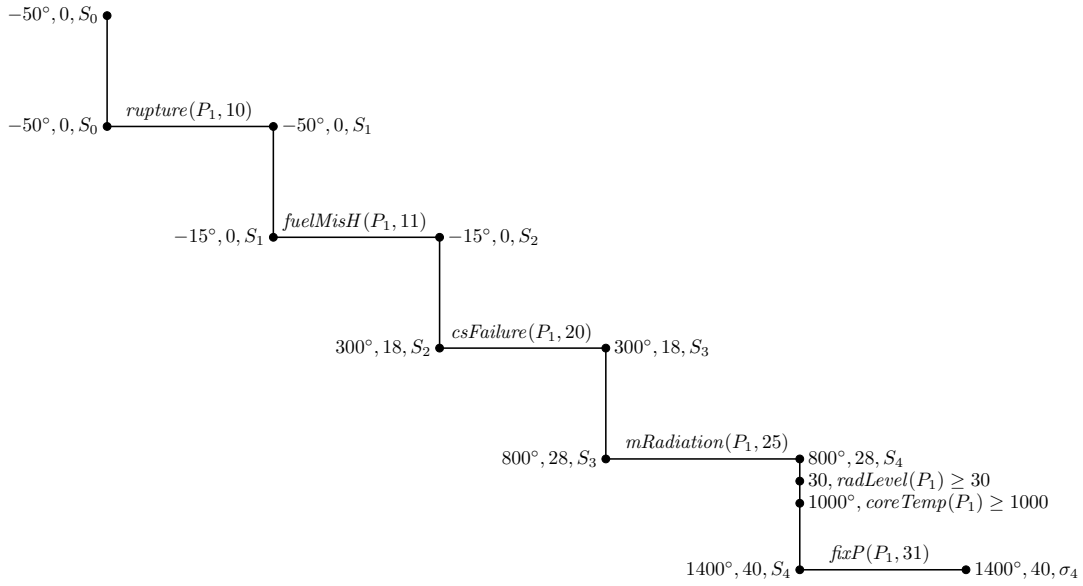

Figure 4.2: Primary Cause in Hybrid Domains: Conjunctive Case

Given this, according to the Definition 4.4.1, the primary cause is as follows:

Proposition 6.

$$\mathcal{D}_{npp2} \models \mathit{CausesDirectly}_{temp}(\mathit{csFailure}(P_1, 20), 2, \varphi_4, \sigma_4).$$

Proof sketch. Using similar arguments as in the proof of Proposition 5, I can show that:

$$\mathcal{D}_{npp2} \models \mathit{CausesDirectly}(\mathit{csFailure}(P_1, 20), 2, \gamma_1, S_4),$$

$$\mathcal{D}_{npp2} \models \mathit{CausesDirectly}(\mathit{fuelMisH}(P_1, 11), 1, \gamma_4, S_4).$$

Also, I can show that:

$$\mathcal{D}_{npp2} \models \mathit{AchvSit}(S_4, \mathit{coreTemp}(P_1) \geq 1000, \sigma_4) \wedge \gamma_1[S_4],$$

$$\mathcal{D}_{npp2} \models \mathit{AchvSit}(S_4, \mathit{radLevel}(P_1) \geq 30, \sigma_4) \wedge \gamma_4[S_4].$$

Since $\mathit{csFailure}(P_1, 20)$, executed at time-stamp 2, is the latest action between the two, it is responsible for the conjunction becoming true, and hence according to Definition 4.4.1, it is the primary cause of the whole conjunction. □

Intuitively, since the direct achievement cause of an effect must be the action that directly brought about the effect, it seems reasonable to call this last relevant action the primary cause of the effect.

4.6.4 Example: Disjunctive Case

For the disjunctive case, let us consider another causal setting $\langle \mathcal{D}_{npp2}, \varphi_5, \sigma_5 \rangle$, where the effect φ_5 and the scenario σ_5 are defined as follows:

$$\varphi_5 = (coreTemp(P_1) \geq 1000) \vee (radLevel(P_1) \geq 30),$$

$$\sigma_5 = do([fuelMisH(P_1, 8), rupture(P_1, 10), csFailure(P_1, 20), mRadiation(P_1, 25), fixP(P_1, 31)], S_0).$$

This is depicted in Figure 4.3, which also shows the core temperature and radiation levels at the beginning and end of each situation.

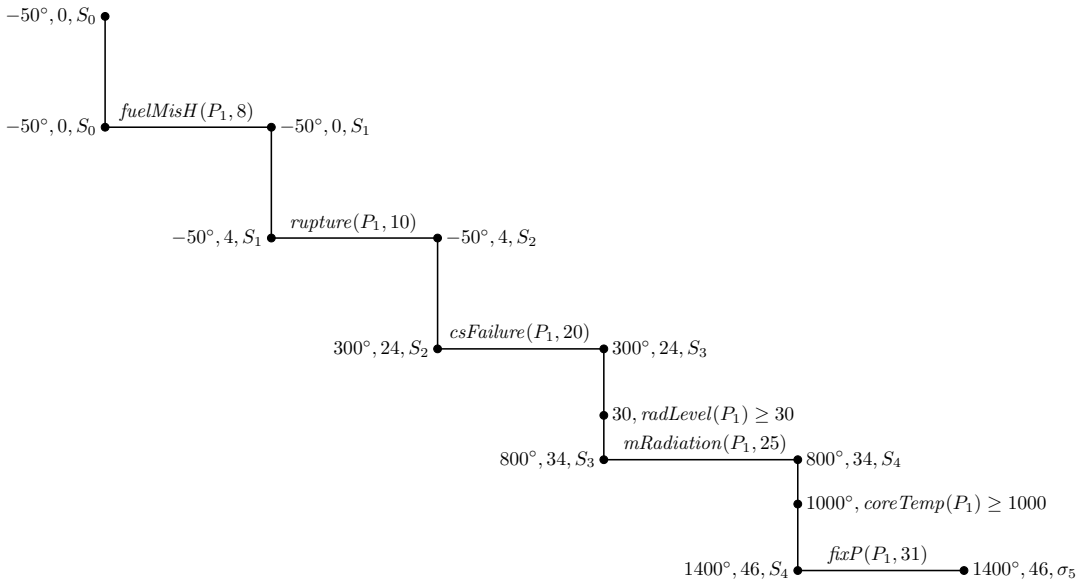

Figure 4.3: Primary Cause in Hybrid Domains: Disjunctive Case

Given this causal setting, according to Definition 4.4.2, we have:

Proposition 7.

$$\mathcal{D}_{npp2} \models \mathit{CausesDirectly}_{\mathit{temp}}(\mathit{fuelMisH}(P_1, 8), 0, \varphi_5, \sigma_5).$$

Proof sketch. Similar to the proof of Proposition 6, I can show that:

$$\mathcal{D}_{npp2} \models \mathit{CausesDirectly}(\mathit{csFailure}(P_1, 20), 2, \gamma_1, S_4),$$

$$\mathcal{D}_{npp2} \models \mathit{CausesDirectly}(\mathit{fuelMisH}(P_1, 8), 0, \gamma_4, S_3).$$

Also, we have:

$$\mathcal{D}_{npp2} \models \mathit{AchvSit}(S_4, \mathit{coreTemp}(P_1) \geq 1000, \sigma_4) \wedge \gamma_1[S_4],$$

$$\mathcal{D}_{npp2} \models \mathit{AchvSit}(S_3, \mathit{radLevel}(P_1) \geq 30, \sigma_4) \wedge \gamma_4[S_3].$$

Even though $\mathit{csFailure}(P_1, 20)$ is the latest contributor towards achieving the effect $\mathit{coreTemp}(P_1) \geq 1000^\circ$ and thus φ_5, it is not the reason for φ_5 becoming true in the given scenario σ_5. According to Definition 4.4.2, since $\mathit{radLevel}(P_1) \geq 30$ has an earlier achievement situation and thus is the reason for the disjunction becoming true in achievement situation S_3, the primary cause of $\mathit{radLevel}(P_1) \geq 30$ is also the primary cause of the entire disjunction φ_5. □

4.6.5 Example: Implicit Primary Cause

Consider the causal setting $\langle \mathcal{D}_{npp3}, \varphi_6, \sigma_6 \rangle$, where the domain theory $\mathcal{D}_{npp3}$, the effect φ_6 and the scenario σ_6 are defined as follows:

$$\mathcal{D}_{npp3} = \mathcal{D}_{npp} \backslash \{\neg \mathit{CSFailed}(P_1, S_0)\} \cup \{\mathit{CSFailed}(P_1, S_0)\},$$

$$\varphi_3 \stackrel{\text{def}}{=} \mathit{coreTemp}(P_1) \geq 1000,$$

$$\sigma_6 = \mathit{do}([\mathit{mRadiation}(P_1, 10), \mathit{fixCS}(P_1, 22), \mathit{mRadiation}(P_1, 25)], S_0).$$

Given this, we have:

Axiom 4.6.5.

$$\mathcal{D}_{npp3} \models \gamma_3[S_0].$$

The scenario is depicted in Figure 4.4.

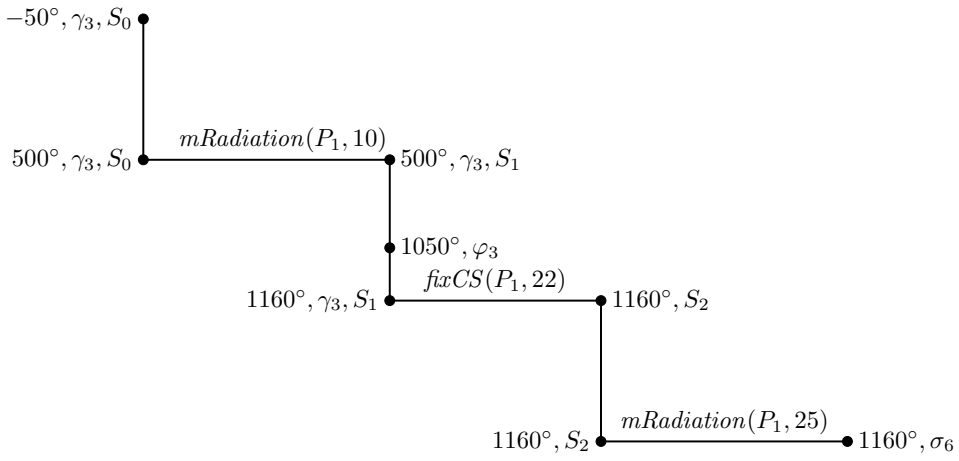

Figure 4.4: Implicit Primary Cause

Given this, I can show that:

Proposition 8.

$$\mathcal{D}_{npp3} \models \neg\exists a, ts.\ CausesDirectly^{prim}_{HTSC}(a, ts, \varphi_3, \sigma_6).$$

Proof. In the initial situation S_0, according to Axiom 4.6.5, the cooling system had already failed, i.e., $CSFailed(P_1, S_0)$. Moreover, by Axiom 3.3.14(a), the pipe of P_1 is not ruptured, and hence the context γ_3 ($\neg Ruptured(P_1) \wedge CSFailed(P_1)$) related to *coreTemp* was already enabled, leading to an increase in *coreTemp* as specified in Axiom 3.3.12. In S_0, after executing the $mRadiation(P_1, 10)$ action, which was executable according to Axiom 3.3.10(e), the effect φ_3 became true in situation $S_1 = do(mRadiation(P_1, 10), S_0)$. According to Axiom 3.3.11(b), $CSFailed(P_1, S_1)$ holds.

Next, according to Axiom 3.3.10(d), the $fixCS(P_1, 22)$ action can be executed in situation S_1. After this action and as specified in Axiom 3.3.11(a) and (b), both $\neg Ruptured(P_1, S_2)$ and $\neg CSFailed(P_1, S_2)$ hold, where $S_2 = do(fixCS(P_1, 22), S_1)$. Furthermore, Axiom 3.3.12 dictates that this action disables context γ_3, which stops the change in *coreTemp* in situation S_2.

Finally, the action $mRadiation(P_1, 25)$ was executable in situation S_2 as per Axiom 3.3.10(e). Since no context was enabled in situation S_2, *coreTemp* remains unchanged

until σ_6, and as a result, the effect φ_3 persists. Therefore, we have:

$$\mathcal{D}_{npp3} \models AchvSit(S_1, \varphi_3, \sigma_6) \wedge \gamma_3[S_1]. \tag{4.25}$$

As per Axiom 4.6.5, γ_3 was already true in S_0 and remained true till the achievement situation S_1. Definition 3.4.6 requires γ_3 to be false before an action can cause it, and hence:

$$\mathcal{D}_{npp3} \models \neg\exists a, ts. CausesDirectly(a, ts, \varphi_3, \sigma_6).$$

□

Thus, no action in σ_6 is directly responsible for the temperature increase contributing to the effect. Here, the primary cause is implicit in the initial situation.

4.7 Conclusion

Building on the foundational work of Batusov and Souchanski [4] as well as Khan and Lespérance [30] in the discrete domains, I have proposed a new definition of primary cause for primitive temporal fluents within a recently developed hybrid temporal version of the situation calculus. First, I defined proper hybrid causal setting where, besides the achievement condition, I also require the effect to be false at the end of the initial situation. I addressed the challenge of pinpointing the exact achievement time point for effects by defining the achievement situation where the effect is achieved, and after which it remains true till the end of the scenario. Instead of using an exact

time point, I use the start and end time points of situations and show how precise identification of causal relationships over time can still be modeled using this.

I explored how one can extend my primary cause definition to settings where the effect is a conjunction or disjunction of two primitive temporal fluents. Next, I studied properties of the primary cause, in particular demonstrated its uniqueness, and examined conditions for its persistence when the scenario changes. Additionally, I showed that in a proper hybrid causal setting, the primary cause might not exist and could be implicit in the initial situation. Finally, I gave detailed examples to illustrate the intuition behind my formalism.

The work presented in this chapter is limited in many ways. For instance, I only dealt with causes of primitive effects and simple conjunctive and disjunctive effects. Also, I focused on primary causes exclusively and ignored secondary or indirect causes. Finally, I only looked at achievement causes, but not maintenance causes. Nonetheless, my proposal shows that formalizing causes even under such strong restrictions is non-trivial and interesting. In the future, I plan to extend this to deal with some of these constraints.

Chapter 5

Another Definition of Primary Cause: A Counterfactual Perspective

5.1 Introduction

Building on my proposal in Chapter 3, where I defined counterfactual situations, analyzed causes in discrete domains, and demonstrated how preempted causes in the absence of actual causes can still produce effects, in this chapter I study how counterfactuals and causes are related in hybrid domains. Khan and Soutchanski [32] previously studied the relationship between actual causes and contributors/contributing actions, and some of this work is motivated by that. To this end, I give another definition of primary cause within the hybrid temporal situation calculus for primitive temporal case and show that this definition is equivalent to the one proposed in Chapter 4 (Definition 4.3.1).

As discussed in Section 3.5, a key challenge is preemption, which complicates the

causal relationship. A particular complication that arises in the context of hybrid temporal situation calculus is that preempted causes can occur even before the direct cause[1]. To address this problem, I first identify the preempted causes/contributors. Subsequently, I introduce the concept of a "defused" situation, where the actual cause along with all preempted contributors are replaced with *noOp* actions, i.e., actions that have no effect. This substitution allows us to isolate the impact of the primary cause by removing the influence of the cause and its preempted contributors. I then show that in this defused situation, either the effect does not occur or the scenario becomes non-executable, as formalized in Theorem 5.3.5. I illustrate this with examples in Section 5.4, showing how temporal ordering affects causal interpretation. The work presented in this chapter is somewhat preliminary in nature.

5.2 Another Definition of Primary Cause

I start by introducing various notions of contributors and define actual cause through contributions in the achievement situation. First, I define *direct possible contributors*, i.e., actions that directly initiate the change in the values of temporal fluents.

Definition 5.2.1 (Direct Possible Contributor)**.** *Given a hybrid temporal action theory $\mathcal{D}$, and an effect φ, which is a constraint on the value of a temporal fluent f, an action α executed in situation s_α is called a direct possible contributor to φ*

[1]Note that this cannot be the case in discrete domains; otherwise, by definition, the preempted cause would have been the actual cause.

(denoted using $DirPossContr(\alpha, s_\alpha, \varphi)$), if and only if the following holds:

$$\mathcal{D} \models \exists i, s_\varphi, \sigma, ts.\ executable(s_\alpha) \wedge Poss(\alpha, s_\alpha) \wedge timeStamp(s_\alpha) = ts \wedge s_\alpha < s_\varphi \leq \sigma$$
$$\wedge \neg\varphi[time(\alpha), s_\alpha] \wedge \varphi[end(s_\varphi, \sigma), s_\varphi] \wedge CausesDirectly(\alpha, ts, \gamma_i^f, s_\varphi).$$

That is, α executed in situation s_α is a direct possible contributor of φ, i.e., $DirPossContr(\alpha, s_\alpha, \varphi)$, if and only if s_α is an executable situation, it is possible to execute α in s_α, the timestamp of s_α is ts, the effect was false in s_α but later became true by the end of a future situation s_φ within some scenario σ, and α executed in timestamp ts is a direct cause of some context γ_i^f of f in situation s_φ. Note that, the last conjunct, i.e., $CausesDirectly(\alpha, ts, \gamma_i^f, s_\varphi)$, ensures that γ_i^f has been true since the execution of α, and thus given that $\neg\varphi[time(\alpha), s_\alpha] \wedge \varphi[end(s_\varphi, \sigma), s_\varphi]$, it must have been the enabled context when φ was achieved. Thus α must have been the action that achieved φ. It should also be noted that φ and γ_i^f are not required to be true in σ. Finally, note that s_φ and σ are not guaranteed to be within the actual scenario.

In the following, I will also use a variant of *DirPossContr* that makes s_φ and σ explicit, i.e., $DirPossContr(\alpha, s_\alpha, s_\varphi, \sigma, \varphi)$.

Next, I define the notion of *direct actual contributors*. These actions are direct possible contributors that are contained within a given causal setting (scenario).

Definition 5.2.2 (Direct Actual Contributor)**.** *Given a hybrid temporal achievement causal setting $\langle \mathcal{D}, \sigma, \varphi \rangle$, an action α executed in situation s_α is called a direct actual*

contributor to an effect φ—which is a constraint on the value of a temporal fluent f, i.e., $DirActContr(\alpha, s_\alpha, s_\varphi, \varphi, \sigma)$, if and only if the following holds:

$$\mathcal{D} \models \exists \sigma'.\ DirPossContr(\alpha, s_\alpha, s_\varphi, \sigma', \varphi) \wedge \sigma' \leq \sigma.$$

That is, α executed in situation s_α is a direct actual contributor to effect φ which was brought about in situation s_φ within scenario σ if and only if α executed in s_α is a direct possible contributor of φ for some scenario σ', s_φ is an achievement situation of φ, and σ' occurs within the scenario σ.

We want to define primary cause as a direct actual contributor such that, after its contribution, the effect is achieved and persists until the end of a scenario—i.e., a direct actual contributor to an active context in the achievement situation. Using this, I now give another definition of primary cause.

Definition 5.2.3 (Primary Cause)**.** *Given a hybrid temporal achievement causal setting, $\langle \mathcal{D}, \sigma, \varphi \rangle$, an action α, executed at time-stamp ts, is called the primary cause of the effect φ—which is a constraint on the value of a temporal fluent f, denoted by $PrimaryCause(\alpha, ts, \varphi, \sigma)$, if and only if*

$$\mathcal{D} \models \exists s_\alpha, s_\varphi.\ AchvSit(s_\varphi, \varphi, \sigma) \wedge timeStamp(s_\alpha) = ts \wedge DirActContr(\alpha, s_\alpha, s_\varphi, \varphi, \sigma).$$

That is, α executed at ts is the primary cause of φ in scenario σ if and only if s_φ is the achievement situation of φ in σ, and α executed in some situation s_α with timestamp

ts is a direct actual contributor of φ in σ.

Now, I prove that this new definition is equivalent to the one proposed in Chapter 4, i.e., $CausesDirectly^{prim}_{temp}(a, ts, \varphi, s)$, in Definition 4.3.1.

Theorem 5.2.1 (Equivalence of Primary Cause and Direct Cause Definitions)**.** *Given a hybrid causal setting $\langle \mathcal{D}, \sigma, \varphi \rangle$, we have:*

$$\mathcal{D} \models \forall \alpha_1, ts_1, \alpha_2, ts_2. CausesDirectly^{prim}_{temp}(\alpha_1, ts_1, \varphi, s) \wedge PrimaryCause(\alpha_2, ts_2, \varphi, \sigma)$$
$$\supset \alpha_1 = \alpha_2 \wedge ts_1 = ts_2.$$

Proof sketch. By Lemma 4.5.2 and Axiom 3.3.4, the achievement situation s_φ and the context γ_i^f enabled is s_φ are the same in both definitions. According to Definition 4.3.1, α_1 executed in ts_1 directly causes γ_i^f in s_φ. In Definition 5.2.3, α_2 executed in ts_2 directly causes the same γ_i^f in s_φ (see Definitions 5.2.1 and 5.2.2). Given the uniqueness of direct causes (see Lemma 4.5.1), α_1 and α_2 must be the same action and their execution time must also be the same. □

5.3 Defused Situation for Counterfactual Analysis

As discussed in the example given in Section 3.5.2, in discrete domains, if we remove the primary cause from the scenario, the effect might still follow due to the presence of preempted actions. The same applies to hybrid domains: even if we remove the direct cause, another context realized by a subsequent action might come

into effect and bring about the effect. Furthermore, in hybrid domains, preempted causes can occur even before the direct cause, as demonstrated in examples later in Section 5.4. This is because the removal of the actual cause (and the context realized by it) might result in another context, brought about by an earlier action persisting, which might bring about the effect eventually. To address this issue, I will use the following approach: after replacing the primary cause with a *noOp* action with an appropriate time argument, I check if another action becomes the primary cause in the new scenario. If a new primary cause emerges, it must have been a preempted action. So I replace this new primary cause with another *noOp* action and repeat this process until no primary causes remain. By doing this, I thus not only remove the primary cause from the scenario but also all preempted causes/actions. I will show later on that in the resultant scenario, either the effect no longer holds, or the scenario itself becomes non-executable, unless some context inherent in the initial situation brings about the effect (see Theorem 5.3.5).

First, let me define preempted contributors, which are actions that could have caused the effect if it were not for the primary cause.

Definition 5.3.1 (Preempted Contributors)**.** *Given a hybrid temporal achievement causal setting, $\langle \mathcal{D}, \sigma, \varphi \rangle$, an action a, time-stamp ts, and situation σ', PreempContr*

$(a, ts, \sigma', \varphi, \sigma)$ *is defined as follows:*

$$
\begin{aligned}
PreempContr(a, ts, \sigma', \varphi, \sigma) \stackrel{\text{def}}{=} \forall P.[\forall a, ts, \varphi, \sigma, \sigma'.(&PrimaryCause(a, ts, \varphi, \sigma) \\
&\wedge CF_{one}(\sigma', \sigma, \langle noOp(time(a)), a, ts\rangle) \supset P(a, ts, \sigma', \varphi, \sigma)) \\
\wedge \forall a', ts', \varphi, \sigma, \sigma'.\ (\exists a'', ts'', \sigma''.(&P(a'', ts'', \sigma'', \varphi, \sigma) \\
&\wedge PrimaryCause(a', ts', \varphi, \sigma'') \wedge CF_{one}(\sigma', \sigma'', \langle noOp(time(a')), a', ts'\rangle) \\
&\supset P(a', ts', \sigma', \varphi, \sigma)) \\
]\ \supset\ &P(a, ts, \sigma', \varphi, \sigma).
\end{aligned}
$$

Thus, *PreempContr* is defined to be the least relation P such that if a executed at time-stamp ts is a primary cause of φ in scenario σ, then $(a, ts, \sigma', \varphi, \sigma)$ is in P, where σ' is a single action counterfactual situation to σ obtained by replacing a at ts with $noOp(time(a))$. And if a'', ts'', σ'', φ, and σ is in P, a' executed at time-step ts' is a primary cause of φ in σ'', and σ' is a single action counterfactual situation of σ'' that can be obtained by replacing a' at ts' with the $noOp(time(a'))$ action, then $(a', ts', \sigma', \varphi, \sigma)$ is also in P.

Here, while we do not check the executability of the updated scenario (e.g. by using $CFEx_{one}$ instead of CF_{one}), this is guaranteed by our definition of primary cause, which requires the scenario to be executable (see Definition 5.2.3). Additionally, note that *PreempContr* returns a set of tuples containing σ', where σ' is a situation obtained by removing the actual cause and zero or more preempted contributors.

We need to identify the tuple containing a situation σ' where *all*[2] the preempted contributors are replaced with *noOp* actions. That is, we are looking for the tuple with the highest number of *noOp* actions. To achieve this, let us define the number of *noOp* actions in σ, denoted by $|\sigma|$.

Definition 5.3.2 ($|\sigma|$)**.**

$$|s| = \begin{cases} 0, & \textit{if} \quad s = S_0, \\ 0 + |s'|, & \textit{if} \quad s = do(a, s') \wedge \neg\exists t.\ a = noOp(t), \\ 1 + |s'|, & \textit{if} \quad \exists t.\ s = do(noOp(t), s'). \end{cases}$$

Using this, I define a defused situation as one that contains the maximum number of *noOp* actions.

Definition 5.3.3 (Defused Situation)**.**

$$\begin{aligned} &DefusedSit(\varphi, \sigma, \sigma') \stackrel{\text{def}}{=} \exists a', ts'.\ PreempContr(a', ts', \sigma', \varphi, \sigma) \\ &\qquad \wedge \forall \sigma'', a'', ts''.\ PreempContr(a'', ts'', \sigma'', \varphi, \sigma) \wedge \sigma' \neq \sigma'' \supset |\sigma''| < |\sigma'|. \end{aligned}$$

That is, σ' is the defused situation of σ with respect to φ if there exists an action α' and a time-stamp ts' such that $(a', ts', \sigma', \varphi, \sigma)$ is in the set of preempted contributors for some a', and for any other σ'' such that $(a'', ts'', \sigma'', \varphi, \sigma)$ is in the set of preempted

[2]In fact we might not be able to remove all preempted contributors, e.g., when the scenario becomes non-executable. So we are really talking about identifying the tuple where the maximum number of preempted causes are removed.

contributors for some a'', ts'', if σ' is different than σ'', then σ' must have more $noOp$ actions in it. Thus a defused situation is one where the actual cause along with the highest number of preempted causes are removed. Note that this might involve removing all the preempted causes. It is also possible that not all preempted contributors were removed, e.g., when replacing an action with $noOp$ made the scenario non-executable (and thus the preempted action is no longer an achievement cause in the modified scenario).

I will next present a theorem that shows using the defused situation that the effect is indeed counterfactually dependent on the primary cause in the sense that if the cause along with most of the preempted actions are removed to obtain a defused situation, either the effect does not follow in this situation, or the scenario becomes non-executable, unless a relevant context was already true in the initial situation S_0. But before presenting the counterfactual dependence theorem, let me introduce some supporting lemmata.

Lemma 5.3.1.

$$\mathcal{D} \models \exists a, ts.\ \mathit{PrimaryCause}(a, ts, \varphi, \sigma) \supset \exists \sigma'.\ \mathit{DefusedSit}(\varphi, \sigma, \sigma').$$

Proof. By Definition 5.2.3 and 5.3.1, if a is a primary cause then σ must have been a non-initial situation (i.e., $\sigma \neq S_0$), and thus we can always construct σ' by replacing a with the $noOp(time(a))$ action in σ. □

Next, I show that two counterfactual situations obtained by replacing the same number of *noOp* actions with preempted contributors, must be the same.

Lemma 5.3.2.

$$\exists a_1, ts_1, \sigma_1, \varphi, \sigma, a_2, ts_2, \sigma_2, \varphi, \sigma.\ PreempContr(a_1, ts_1, \sigma_1, \varphi, \sigma)$$
$$\wedge\ PreempContr(a_2, ts_2, \sigma_2, \varphi, \sigma) \wedge |\sigma_1| = |\sigma_2| \supset \sigma_1 = \sigma_2.$$

Proof sketch (by induction). I start by induction on $|\sigma_1|$, the number of *noOp* actions in σ_1. For $n = 1$, both σ_1^b and σ_2^b contain exactly one *noOp* action, which corresponds to a substitution of the primary cause of φ in σ. By Definition 5.3.1, the Property 4.5.3 (the uniqueness of the primary cause), and Theorem 5.2.1 (that the two definitions of primary causes are equivalent), we have:

$$PreempContr(a_1, ts_1, \sigma_1^b, \varphi, \sigma) \wedge PreempContr(a_2, ts_2, \sigma_2^b, \varphi, \sigma) \supset \sigma_1^b = \sigma_2^b.$$

Assume that the consequence holds for $|\sigma_1^k| = |\sigma_2^k| = k$.
We will show that it holds for $|\sigma_1| = |\sigma_2| = k + 1$. By the same argument as in the base case (i.e., that replacing the actual cause from the same situation yields a unique situation due to the uniqueness of primary cause), it can be shown that $\sigma_1 = \sigma_2$. □

Using this, I can show that:

Lemma 5.3.3.

$$\exists \sigma', \sigma''.\ DefusedSit(\varphi, \sigma, \sigma') \wedge DefusedSit(\varphi, \sigma, \sigma'') \supset \sigma' = \sigma''.$$

Proof. Follows directly from Lemma 5.3.2, Definition 5.3.3, and the fact that since the scenario σ is finite, only a finite number of actions could be replaced with *noOp* actions. □

I also show that in defused situation, no primary cause exists:

Lemma 5.3.4.

$$DefusedSit(\varphi, s, s') \supset \neg \exists b, ts_b. PrimaryCause(b, ts_b, \varphi, s').$$

Proof (by contradiction). Fix φ_1, s_1, s_1' and assume on the contrary that for some b_1 and ts_{b_1},

$$PrimaryCause(b_1, ts_{b_1}, \varphi_1, s_1').$$

According to Definition 5.3.1,

$$PreempContr(b_1, ts_{b_1}, s_1'', \varphi, s_1'),$$

where s_1'' is a counterfactual situation of s_1' with b_1 replaced by $noOp(time(b_1))$. This implies $|s_1'| < |s_1''|$, contradicting Definition 5.3.3 and Lemma 5.3.3, which requires s_1' to be the unique defused situation and s_1' to have the maximum number of *noOp*

actions. Therefore, no such b_1 exists, proving the lemma. □

Theorem 5.3.5 (Counterfactual Dependence)**.** *Given a hybrid temporal causal setting, $\langle \mathcal{D}, \sigma, \varphi \rangle$, the following holds:*

$$\exists a, ts. PrimaryCause(a, ts, \varphi, s) \supset$$
$$(\exists s'.\ DefusedSit(\varphi, s, s') \wedge$$
$$\wedge (\bigwedge_i \neg\gamma_i^f[S_0] \supset \neg(\varphi[start(s'), s'] \wedge Executable(s')))).$$

This states that if there is an action a and timestamp ts such that a executed at ts is a primary cause of effect φ in scenario s, then there is a defused situation s' relative to φ and s, and if additionally it is known that all the contexts of φ are inactive initially in S_0 (i.e., $\bigwedge_i \neg\gamma_i^f[S_0]$), then it must be the case that φ is false in the defused situation s', unless s' has become non-executable.

Proof sketch. Fix $A_1, ts_1, \varphi_1, s_1$ and assume that $PrimaryCause(A_1, ts_1, \varphi_1, s_1)$.

From Lemma 5.3.1, it follows that there is a situation, say s_1', such that $DefusedSit(\varphi_1, s_1, s_1')$.

Now, assume that $\bigwedge_i \neg\gamma_i^f[S_0]$. Also assume that $Executable(s_1')$. We need to show that $\neg\varphi_1[start(s_1'), s_1']$. I will prove this by contradiction. Assume that $\varphi_1[start(s_1'), s_1']$. Since all the contexts of the only fluent f in φ_1 were initially false, there must be an action $A_1'(t)$ executed at some timestamp ts' that brought about some context of φ which eventually achieved φ. Consider the situation that replaces $A_1'(t)$ with $noOp(t)$

in s_1', let's call it s_1^*. By Definition 5.3.1, s_1^* must be a preempted contributor, i.e., $PreempContr(A_1'(t), ts', s_1^*, \varphi, \sigma)$.

Moreover, $|s_1^*| > |s_1'|$; but this means that by Definition 5.3.3, s_1' cannot be the defused situation with respect to φ and s_1, which is contradictory to the above assumption. □

Note that the above analysis is not meant to be a proof of correctness. For instance, if one were to define actual cause as the last non-$noOp$ action in the scenario, the property should still follow. Instead, the above property shows an intuitively justifiable property of causes, that if one removes causes and preempted actions from the scenario, under certain reasonable and intuitive conditions (i.e., that no context of the temporal fluent holds initially), the effect will disappear.

5.4 Examples: Counterfactual Analysis

In this section, I present three examples illustrating different counterfactual scenarios: (i) a scenario that is executable but where the effect does not hold, (ii) a scenario that is non-executable and where the effect does not hold, and (iii) a scenario that is non-executable but where the effect holds.

5.4.1 Example 1

Let us revisit our Example 4.6.1, where in the causal setting $\langle \mathcal{D}_{npp}, \varphi_3, \sigma_3 \rangle$, we have:

$$\varphi_3 = coreTemp(P_1) \geq 1000,$$

$$\sigma_3 = do([rupture(P_1, 5), csFailure(P_1, 15), mRadiation(P_1, 20), fixP(P_1, 26)], S_0),$$

$$\mathcal{D}_{npp} \models CausesDirectly^{prim}_{temp}(csFailure(P_1, 15), 1, \varphi_3, \sigma_3).$$

According to Definition 5.3.1, I replace the primary cause of φ_3 with the $noOp(15)$ action, resulting in the scenario σ'_3, i.e., $CF_{one}(\sigma_3, \sigma'_3, \langle noOp(15), csFailure(P_1), 1 \rangle)$.

$$\sigma'_3 = do([rupture(P_1, 5), noOp(15), mRadiation(P_1, 20), fixP(P_1, 26)], S_0).$$

σ'_3 is illustrated in Figure 5.1.

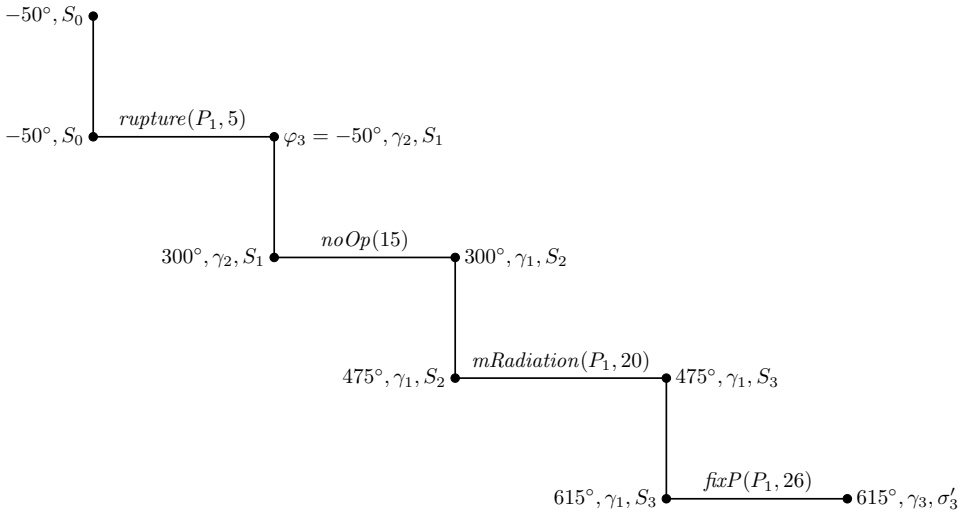

Figure 5.1: Example 1. Counterfactuals in Primitive Temporal Case

By Axiom 3.3.14(a) and (b), and given that none of the contexts $\gamma_1, \gamma_2, \gamma_3$ were active in S_0, Axiom 3.3.12 and 3.3.14(c) imply that the temperature of P_1 remains at $-50°$ in S_0 at time 5. In situation S_0, the $rupture(P_1, 5)$ action is executed. According to Axiom 3.3.10(a), this action was possible to execute in S_0. As stated in Axiom 3.3.14(b) and 3.3.11(b), we have $\neg CSFailed(do(rupture(P_1, 5), S_0))$. Therefore, in this situation, the context γ_2 is true. According to Axiom 3.3.12, this initiates the increase in *coreTemp* in accordance with δ_2, meaning the temperature rises by $35°$ per second. Consequently, the core temperature reaches $300°$ at time 15.

In situation $S_1 = do(rupture(P_1, 5), S_0)$, an always-possible action $noOp(15)$ is executed at time 15, allowing *coreTemp* to continue increasing while the context γ_2 remains true. The *coreTemp* reaches $475°$ at time 20.

Following the execution of the $mRadiation(P_1, 20)$ action in situation $S_2 = do(noOp(15), S_1)$, which was also executable according to Axiom 3.3.10(e), $Ruptured(P_1, S_3)$ continues to hold, as stated in Axiom 3.3.11(a), where $S_3 = do(mRadiation(P_1, 20), S_2)$. The context γ_2 remains unchanged, as does the rate of change. The *coreTemp* continues to rise and reaches $615°$ at time 26.

Finally, $fixP(P_1, 26)$ is executed which is also possible to execute in S_4 as per Axiom 3.3.10(b), and given $Ruptured(P_1, S_3)$.

Proposition 9.

$$\mathcal{D}_{npp} \models \neg\varphi_2[start(\sigma_3'), \sigma_3'].$$

Initially, the core temperature was $-50°$, and by the end of the scenario, it reached

615°, failing to satisfy the achievement condition. Hence, the scenario σ_3' is executable, but the effect does not hold in σ_3', i.e., $\neg\varphi_3[start(\sigma_3'), \sigma_3']$.

5.4.2 Example 2

Consider a causal setting $\langle \mathcal{D}_{npp}, \varphi_3, \sigma_7 \rangle$, where the effect φ_3 and the scenario σ_7 are defined as follows:

$\varphi_3 = coreTemp(P_1) \geq 1000,$

$\sigma_7 = do([rupture(P_1, 5), csFailure(P_1, 15), mRadiation(P_1, 26), fixP(P_1, 40)], S_0).$

This is depicted in Figure 5.2.

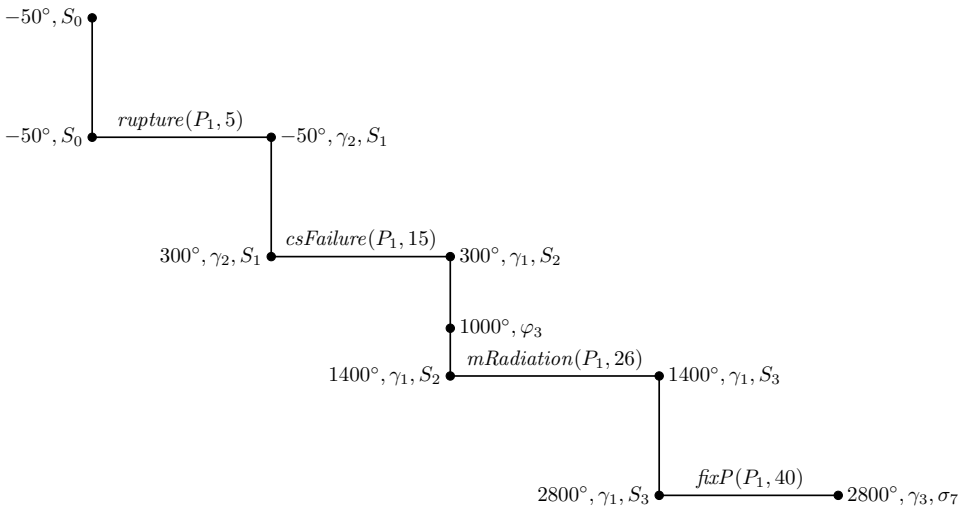

Figure 5.2: Example 2. Figure 1/2. Counterfactuals in HTSC

Using similar arguments as given in Example 5.4.1 and according to Definition

4.3.1, I can show the following result about direct cause.

$$\mathcal{D}_{npp} \models \mathit{CausesDirectly}^{prim}_{temp}(\mathit{csFailure}(P_1, 15), 1, \varphi_3, \sigma_7).$$

We now have σ_7', where the primary cause $\mathit{csFailure}(P_1, 15)$ has been substituted with $\mathit{noOp}(15)$, i.e., $CF_{one}(\sigma_7, \sigma_7', \langle \mathit{noOp}(15), \mathit{csFailure}(P_1), 1\rangle)$.

$$\sigma_7' = do([\mathit{rupture}(P_1, 5), \mathit{noOp}(15), \mathit{mRadiation}(P_1, 26), \mathit{fixP}(P_1, 40)], S_0).$$

σ_7' is illustrated in Figure 5.3.

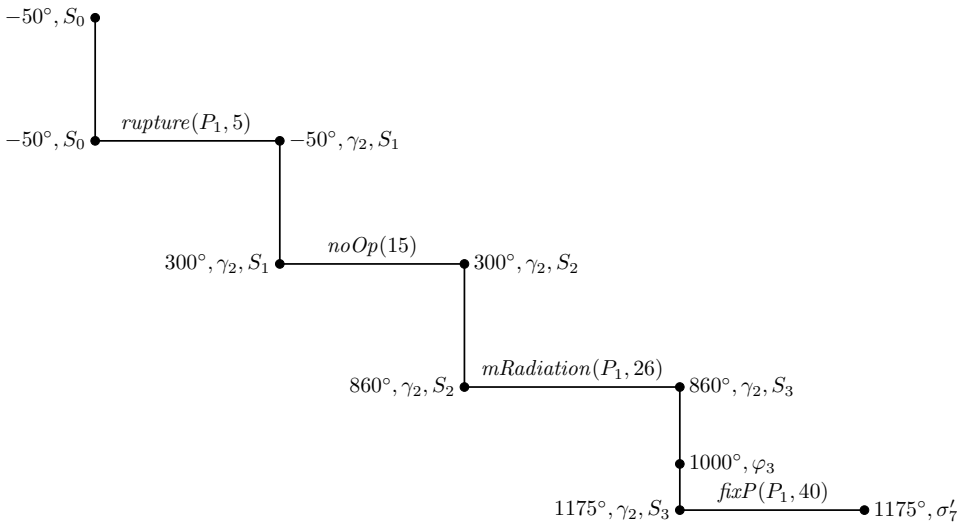

Figure 5.3: Example 2. Figure 2/2. Counterfactuals in HTSC

By following the same reasoning as before and as shown in Figure 5.3, the following conclusion is easily seen:

$$\mathcal{D}_{npp} \models \mathit{CausesDirectly}^{prim}_{temp}(\mathit{rupture}(P_1, 5), 1, \varphi_3, \sigma_7').$$

Now I replace $rupture(P_1, 5)$ with $noOp(5)$.

$$\sigma_7'' = do([noOp(5), noOp(15), mRadiation(P_1, 26), fixP(P_1, 40)], S_0).$$

I then derive the following results in σ_7'':

Proposition 10.

$$\mathcal{D}_{npp} \models \neg\varphi_3[start(\sigma_7''), \sigma_7''] \wedge \neg Executable(\sigma_7'').$$

The scenario σ_7'' is non-executable because the action $fixP(P_1, 40)$ cannot be executed in situation S_3. According to the initial state axiom 3.3.14(a), $Ruptured(P_1, S_0)$ was false, and during the scenario, no pipe rupture action (e.g., $rupture(P_1, t)$) was performed to make the precondition $Ruptured(P_1, S_3)$ true as required by Axiom 3.3.10(a). Additionally, no action in the scenario enabled a context for *coreTemp*, so its value remains the same as in the initial situation, which was $-50°$ (see Axiom 3.3.12).

5.4.3 Example 3

Consider a causal setting $\langle \mathcal{D}_{npp}, \varphi_3, \sigma_8 \rangle$, where the effect φ_3 and the scenario σ_8 are defined as follows:

$\varphi_3 = coreTemp(P_1) \geq 1000,$

$\sigma_8 = do([rupture(P_1, 5), csFailure(P_1, 15), mRadiation(P_1, 26), fixCS(P_1, 40)], S_0).$

This is depicted in Figure 5.4.

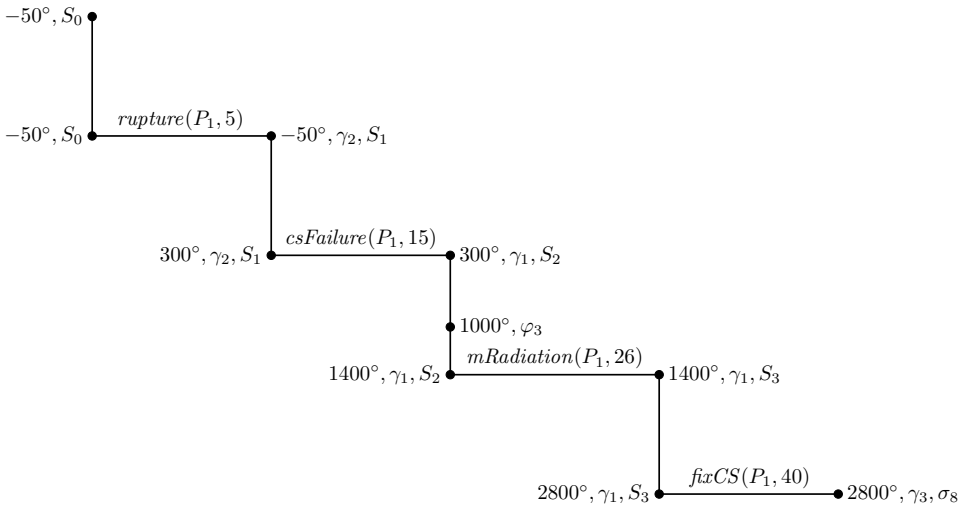

Figure 5.4: Example 3. Figure 1/2. Counterfactuals in HTSC

As illustrated in Figure 5.4, we have the following result about achievement of φ_3 in σ_8:

$$\mathcal{D}_{npp} \models CausesDirectly^{prim}_{temp}(csFailure(P_1, 15), 1, \varphi_3, \sigma_8).$$

Now we have σ_8' such that the primary cause is replaced with $noOp(15)$, and is

depicted in Figure 5.5.

$$\sigma_8' = do([rupture(P_1, 5), noOp(15), mRadiation(P_1, 26), fixCS(P_1, 40)], S_0).$$

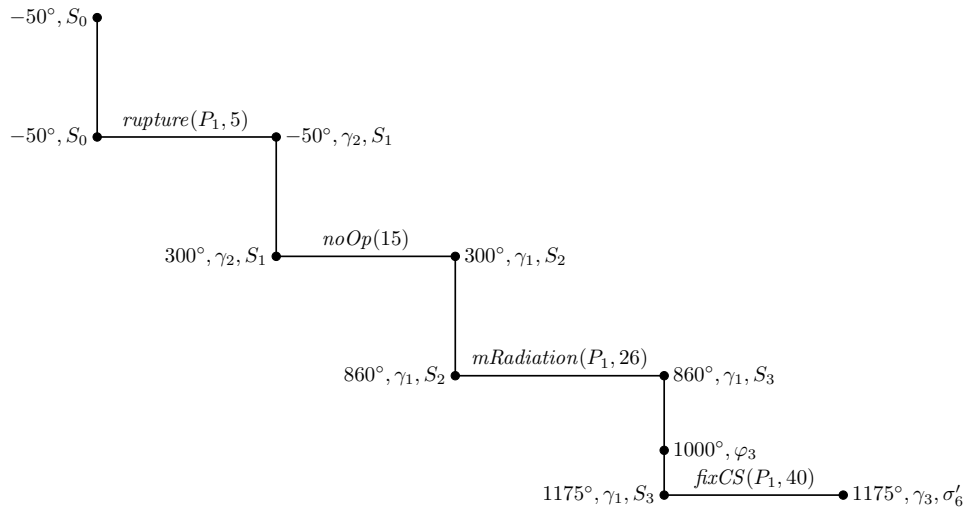

Figure 5.5: Example 3. Figure 2/2. Counterfactuals in HTSC

We have the following direct cause of φ_3 in scenario σ_8':

$$\mathcal{D}_{npp} \models CausesDirectly(rupture(P_1, 5), 0, \gamma_1, S_3).$$

Proposition 11.

$$\mathcal{D}_{npp} \models \neg Executable(\sigma_8').$$

Even though the effect is achieved in the scenario σ_8', the scenario σ_8' itself has become non-executable because the action $fixCS(P_1, 40)$ cannot be executed as per Axiom

3.3.10(d). Per the initial state axiom 3.3.14(b), $CSFailure(P_1, S_0)$ was false, and during the scenario, no cooling system failure action (e.g., $csFailure(P_1, t)$) was performed to satisfy the precondition $CSFailure(P_1, S_3)$, as required by Axiom 3.3.10(c).

5.5 Conclusion

In this chapter, I extended my counterfactual analysis in discrete domains to temporal cases. I first introduced a new definition of primary cause based on the notion of contributions. Note that while I defined primary cause as direct actual contributor of φ in the achievement situation s_φ, it would have been more intuitive to rather define it as the lastest/last direct actual contributor of φ. I leave this for future work. I showed that the two definitions of primary cause (Definition 4.3.1 and 5.2.3) are equivalent. To formally study the contribution of the primary cause on the effect formula and to address the preemption issue, I identified and replaced all preempted contributors/causes with *noOp* actions before verifying if the effect still follows. This led to Theorem 5.3.5, which states that if all preempted causes are replaced with *noOp* actions, then either the effect does not occur or the scenario itself becomes non-executable.

Through illustrative examples, I demonstrated how the temporal ordering of events impacts causal reasoning and highlighted the necessity of addressing preempted actions to establish causality accurately. Overall, this study contributes to our ability

to accurately determine causes in hybrid domains by isolating and removing the influence of preempted causes. In the future, I plan to capture a more intuitive definition of causes via contributions (as discussed in Section 5.3), and extend this analysis to cover compound effects as well as indirect causes.

Chapter 6

Conclusions and Future Research

6.1 Contributions

Motivated by the hybrid nature of change in the real world, in this thesis, I studied actual causes in the hybrid temporal situation calculus. I investigated causation through two perspectives: the foundational approach, where I studied actions and their effects to define achievement causation, and the counterfactual approach, where I defined causation through contributions and justified it using a counterfactual analysis. I started by examining some popular existing approaches and their limitations. Following this, I discussed a foundational framework for modeling dynamic domains, i.e., hybrid temporal situation calculus. Building on prior work in discrete domains, I proposed a formalization of the primary achievement cause in the hybrid temporal situation calculus. To the best of my knowledge, this is the first attempt to address causation within hybrid temporal action-theoretic frameworks. The only other formal effort in this area is the recent work by Halpern and Peters [23], that was discussed

in Chapter 2. However, their framework lacks grounding in a proper action theory and consequently suffers from significant expressive limitations. Below, I will outline the contributions of this thesis.

1. Counterfactual Scenarios in the Situation Calculus

In Chapter 3, I examined counterfactual scenarios, specifically what would have happened if certain actions had not been executed in the given scenario. I formalized both single-action and multiple-action counterfactual situations along with their executable variants. Additionally, I demonstrated how actual causes in discrete domains (Definition 3.4.7) encounter issues with preemption, where the effect might still follow due to preempted causes; see Theorem 3.5.2.

2. A Novel Foundational Definition of Primary Achievement Cause in Hybrid Dynamic Domains

In Chapter 4, I first defined a proper hybrid causal setting (Definition 4.2.1). I then proposed a definition of the primary achievement cause for effects that are constraints on the values of primitive temporal fluents (Definition 4.3.1) and defined the achievement situation of the effect (Definition 4.3.3) to ensure consistency with the temporal ordering of events. In this, I focused on primitive temporal fluents and direct causes exclusively. Additionally, I conducted preliminary work to address conjunctive and disjunctive cases of temporal fluents as effects (Definitions 4.4.1 and 4.4.2). In Section 4.5, I outlined the properties of my primary cause definition, demonstrating

the uniqueness of the primary cause and the conditions under which it persists. I also showed that if change has already been initiated before any action is executed, the effect can occur without apparent cause in a given trace. Finally, I provided examples to illustrate the underlying intuition.

3. Counterfactual Analysis of Primary Cause in Hybrid Domains

In Chapter 5, I extended my notion of counterfactual scenarios from discrete domains to hybrid domains. I then introduced an alternative definition of the primary cause that is based on contributions made by actions to effects (Definition 5.2.3) and demonstrated its equivalence (see Theorem 5.2.1) to my previous definition (Definition 4.3.1). I addressed the preemption issue by defining a defused situation where the effect of the primary cause is isolated from preempted contributors (see Definition 5.3.3). This is done by removing primary as well as any preempted contributors from the scenario to obtain a defused situation and evaluating the effect of cause in this situation. I showed that, in an executable defused situation the effect can no longer be observed (Theorem 5.3.5). Finally, I provided comprehensive examples to illustrate my proposal.

6.2 Conclusion and Future Work

A striking aspect of intelligent reasoning is that it is conditioned by our understanding of causal relationships of actions and their effects. Change in the real world

can be both discrete and continuous, i.e., hybrid. Yet, almost all of the work on formalizing actual causes that can be found in the literature considers change to be discrete. In this thesis, I presented the first account of actual cause in hybrid dynamic domains.

A word about implementablity of this theory: Reiter [50] demonstrated that within the situation calculus one can use regression to solve the projection problem, which reduces projection to entailment from a first-order theory. Batusov, De Giacomo, and Soutchanski also proposed a notion of regression in the hybrid temporal situation calculus [1, 2], which also reduces reasoning to first-order logic. Previously Khan and Soutchanski [32] reported a Prolog implementation of Batusov and Soutchanski's [4] original proposal, which I think can be extended to deal with the hybrid case with some effort. I leave this for future work.

My current proposal is nonetheless limited in several ways. For instance, I only addressed primitive fluents as effects and also did not consider indirect causes. However, this attempt demonstrates that determining causes requires careful modeling and reasoning in hybrid domains, even under strong restrictions. In the future, I plan to identify direct causes for arbitrary fluents, both discrete and temporal. This is challenging, as different fluents have different achievement situations, altering the context-achievement scenario significantly. I also aim to extend this work to discover indirect or secondary causes. This should be achievable along similar lines as in [4, 30], potentially with the aid of the newly proposed regression operator in hybrid temporal situation calculus [1].

Moreover, one could extend my work on counterfactual analysis in temporal cases to discrete cases by following the concept of a defused situation where preempted contributors are removed. To fully capture real-world scenarios, future research could also explore causation in hybrid domains where the effects of actions are non-deterministic, the scenario is non-linear, and epistemic and conative aspects are considered. Finally, it would be interesting to see how this research can be applied to deal with real-world problems, for instance, to detect causes of historical as well as hypothetical faults in dynamic systems such as nuclear power plants or for attributing responsibility and blame in tort law, to name a few.

References

[1] Vitaliy Batusov, Giuseppe De Giacomo, and Mikhail Soutchanski. Hybrid temporal situation calculus. In Marie-Jean Meurs and Frank Rudzicz, editors, *Advances in Artificial Intelligence - 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, Kingston, ON, Canada, May 28-31, 2019, Proceedings*, volume 11489 of *Lecture Notes in Computer Science*, pages 173–185. Springer, 2019.

[2] Vitaliy Batusov, Giuseppe De Giacomo, and Mikhail Soutchanski. Hybrid temporal situation calculus. In Chih-Cheng Hung and George A. Papadopoulos, editors, *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*, pages 1162–1164. ACM, 2019.

[3] Vitaliy Batusov and Mikhail Soutchanski. Situation calculus semantics for actual causality. In Andrew S. Gordon, Rob Miller, and György Turán, editors, *Proceedings of the Thirteenth International Symposium on Commonsense Reasoning, COMMONSENSE 2017, London, UK, November 6-8, 2017*, volume 2052 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.

[4] Vitaliy Batusov and Mikhail Soutchanski. Situation calculus semantics for actual causality. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1744–1752. AAAI Press, 2018.

[5] Alexander Bochman. Actual causality in a logical setting. In *Proceedings of the*

Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pages 1730–1736. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[6] Craig Boutilier, Raymond Reiter, Mikhail Soutchanski, and Sebastian Thrun. Decision-theoretic, high-level agent programming in the situation calculus. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pages 355–362, 2000.

[7] Ernest Davis, Andrew Haas, and Lenhart K. Schubert. Monotonic solution of the frame problem in the situation calculus. In H.E. Kyburg, R.P. Loui, and G.N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 97–155. Kluwer Academic Publishers, 1990.

[8] Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque. Congolog, a concurrent programming language based on the situation calculus. In *Artificial Intelligence*, volume 121, pages 109–169. Elsevier, 2000.

[9] Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque. Indigolog: A high-level programming language for embedded reasoning agents. In *Multi-Agent Programming: Languages, Platforms and Applications*, pages 31–72. Springer, 2004.

[10] Giuseppe De Giacomo and Yves Lespérance. The Nondeterministic Situation Calculus. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, pages 216–226, 11 2021.

[11] Thomas Eiter and Thomas Lukasiewicz. Complexity results for structure-based causality. *Artificial Intelligence*, 142(1):53–89, 2002.

[12] Ari Fogel. On the use of epistemic ordering functions as decision criteria for automated and assisted belief revision in sneps. 2011.

[13] Hojjat Ghaderi, Hector Levesque, and Yves Lespérance. Towards a logical theory of coordination and joint ability. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '07, New York, NY, USA, 2007. Association for Computing Machinery.

[14] Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque. Congolog, a concurrent programming language based on the situation calculus. *Artificial Intelligence*, 121(1-2):109–169, 2000.

[15] Giuseppe De Giacomo, Yves Lespérance, and Adrian R. Pearce. Situation calculus game structures and gdl. In *Proceedings of the Twenty-Second European Conference on Artificial Intelligence*, ECAI'16, page 408–416, NLD, 2016. IOS Press.

[16] Clark Glymour, David Danks, Bruce Glymour, Frederick Eberhardt, Joseph D. Ramsey, Richard Scheines, Peter Spirtes, Choh Man Teng, and Jiji Zhang. Actual causation: A stone soup essay. *Synthese*, 175(2):169–192, 2010.

[17] Andrew R. Haas. The case for domain-specific frame axioms. In Frank M. Brown, editor, *The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop*, pages 343–348. Morgan Kaufmann Publishing, 1987.

[18] Joseph Y. Halpern. Axiomatizing causal reasoning. *Journal of Artificial Intelligence Research*, 12:317–337, 2000.

[19] Joseph Y. Halpern. A modification of the halpern-pearl definition of causality. In Qiang Yang and Michael J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 3022–3033. AAAI Press, 2015.

[20] Joseph Y. Halpern. *Actual Causality*. MIT Press, 2016.

[21] Joseph Y. Halpern and Judea Pearl. Causes and explanations: a structural-model approach: part i: causes. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 194–202, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[22] Joseph Y. Halpern and Judea Pearl. Causes and explanations: A structural-model approach. part i: Causes. *The British Journal for the Philosophy of Science*, 56(4):843–887, 2005.

[23] 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.

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

[25] Mark Hopkins and Judea Pearl. Causality and counterfactuals in the situation calculus. *Journal of Logic and Computation*, 17(5):939–953, 2007.

[26] Mark Hopkins and Judea Pearl. Causality and counterfactuals in the situation calculus. *Journal of Logic and Computation*, 17(5):939–953, 2007.

[27] David Hume. *An Enquiry Concerning Human Understanding*. 1748.

[28] Shakil M. Khan and Yves Lespérance. A logical framework for prioritized goal change. In *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3*, pages 283–290, 2010.

[29] Shakil M. Khan and Yves Lespérance. Sr-apl: a model for a programming language for rational bdi agents with prioritized goals. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '11, page 1251–1252, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[30] Shakil M. Khan and Yves Lespérance. Knowing why - on the dynamics of knowledge about actual causes in the situation calculus. In Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé, editors, *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, pages 701–709. ACM, 2021.

[31] Shakil M. Khan and Maryam Rostamigiv. On explaining agent behaviour via root cause analysis: A formal account grounded in theory of mind. In Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu, editors, *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), volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 1239–1247. IOS Press, 2023.

[32] Shakil M. Khan and Mikhail Soutchanski. Necessary and sufficient conditions for actual root causes. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *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)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 800–808. IOS Press, 2020.

[33] Wolf Kohn, Anil Nerode, and Jeffrey B Remmel. Continualization: A hybrid systems control technique for computing. In *Symposium on control, optimization and supervision (Lille, July 9-12, 1996)*, pages 507–511, 1996.

[34] Florian Leitner-Fischer and Stefan Leue. Causality checking for complex system models. In Roberto Giacobazzi, Josh Berdine, and Isabella Mastroeni, editors, *Verification, Model Checking, and Abstract Interpretation, 14th International Conference, VMCAI 2013, Rome, Italy, January 20-22, 2013. Proceedings*, volume 7737 of *Lecture Notes in Computer Science*, pages 248–267. Springer, 2013.

[35] Hector J. Levesque, Fiora Pirri, and Raymond Reiter. Foundations for the situation calculus. *Electronic Transactions on Artificial Intelligence (ETAI)*, 2:159–178, 1998.

[36] Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl. Golog: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31(1-3):59–83, 1997.

[37] David Lewis. *Counterfactuals*. Harvard University Press, Cambridge, MA, 1973.

[38] John Leslie Mackie. Causes and conditions. *American Philosophical Quarterly*, 2(4):245–264, 1965.

[39] Norman McCain and Hudson Turner. Causal theories of action and change.

In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, page 460–465. AAAI Press, 1997.

[40] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.

[41] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.

[42] Asim Mehmood and Shakil M. Khan. Towards a Definition of Primary Cause in Hybrid Dynamic Domains. *Proceedings of the Canadian Conference on Artificial Intelligence*, may 27 2024. https://caiac.pubpub.org/pub/wr72f8ae.

[43] Anil Nerode. Logic and control. In S. Barry Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *Computation and Logic in the Real World*, pages 585–597, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[44] Charles L. Ortiz Jr. Explanatory update theory: Applications of counterfactual reasoning to causation. *Artificial Intelligence*, 108(1):125–178, 1999.

[45] Judea Pearl. On the definition of actual cause. Technical Report R-259, University of California Los Angeles, 1998.

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

[47] E. P. D. Pednault. ADL and the state-transition model of action. *Journal of Logic and Computation*, 4(5):467–512, 1994.

[48] Fiora Pirri and Ray Reiter. Some contributions to the metatheory of the situation calculus. *J. ACM*, 46(3):325–361, May 1999.

[49] David L. Poole. A framework for decision-theoretic planning i: Combining the situation calculus, conditional plans, probability and utility, 2013.

[50] Raymond Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In *Artificial*

Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, pages 359–380. Academic Press, 1991.

[51] Raymond Reiter. Natural actions, concurrency and continuous time in the situation calculus. In *International Conference on Principles of Knowledge Representation and Reasoning*, 1996.

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

[53] Richard B. Scherl and Hector J. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144(1-2):1–39, 2003.

[54] L. K. Schubert. Monotonic solution of the frame problem in the situation calculus: An efficient method for worlds with fully specified actions. In H. E. Kyburg, R. P. Loui, and G. N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 23–67. Kluwer Academic Press, Boston, MA, USA, 1990.

[55] Herbert A. Simon. Causal ordering and identifiability. *Models of Discovery. Boston Studies in the Philosophy of Science*, 54, 1977.

[56] Mikhail Soutchanski. Execution monitoring of high-level temporal programs. 07 1999.

[57] Richard W. Wright. Causation in tort law. *California Law Review*, 73(6):1735, 1985.

[58] 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. *AI Soc.*, 38(4):1453–1464, 2023.