The 37th Canadian Conference on Artificial Intelligence
DOI: 0

Canadian AI 2024
27—31 May 2024, Guelph, Ontario

Towards a Definition of Primary Cause in Hybrid Dynamic Domains

Asim Mehmood, Shakil M. Khan*
Department of Computer Science
University of Regina, Saskatchewan, Canada

Abstract

The notion of primary cause of an effect or outcome can be defined as the immediate reason behind the outcome, representing the phenomenon –action or event in a dynamic framework– that is directly responsible for bringing about the effect. When causal analysis is done relative to an observed effect and an observed history of actions or events, this is further categorised as actual or token-level primary cause. While there has been a lot of work on actual primary as well as indirect causes in discrete dynamic domains, very few studies address causation in hybrid dynamic domains, that is dynamic systems where change can be both discrete and continuous. Building on recent progress, in this paper we propose a first definition of primary cause in a hybrid action-theoretic framework. Our proposal is set within a hybrid variant of the situation calculus. We show that our formalization has some basic intuitive properties.

Keywords: Actual cause, Hybrid dynamic systems, Situation calculus, Logic.

1. Introduction

A fundamental task in dynamic domains is to figure out the causes of change, e.g. that of an observed effect becoming true given a history of actions, a problem known as actual or token-level causation. Based on Pearl's original work [1, 2], Halpern and Pearl and others [3–9] have extensively studied the problem of actual causation and significantly advanced this field. Halpern and Pearl's approach is based on the concept of structural equation models (SEM) [10] and follows the Humean counterfactual definition of causation. The latter states that "an outcome B is caused by an event A" is the same as saying that "had A never occurred, B never had existed". This definition suffers from the problem of preemption: it could be the case that in the absence of event A, B would still have occurred due to another event, which in the original trace was preempted by A. Halpern and Pearl avoid preemption by performing selective counterfactual analysis and suspending some of the model's mechanisms. While their inspirational work has been used for practical applications, their approach based on SEM has been nevertheless criticized for its limited expressiveness [6, 7, 11], and researchers have attempted to extend it with additional features [12].

In recent years, researchers have become increasingly interested in studying causation within more expressive action-theoretic frameworks, in particular in that of the situation calculus [13–15]. Among other things, this allows one to formalize causation from the perspective of individual agents by defining a notion of epistemic causation [16] and by supporting causal reasoning about conative effects, which in turn has proven to be useful for explaining agent behaviour using causal analysis [17] as well as has the potential for defining important concepts such as responsibility and blame [18].

While there has been much work on actual causality, the vast majority of the work in this area has focused on defining causes in discrete domains. However, a distinguishing feature of the real world is that change can be both discrete and continuous. Unfortunately, very few studies address causation in hybrid dynamic systems, and those that do are framed within

*shakil.khan@uregina.ca

causal models that are known to have limited expressiveness and suffer from a variety of problems. For instance, in [19], Halpern and Peters proposed an extension of SEMs, termed generalized structural-equations models (GSEMs). GSEMs can capture hybrid systems by allowing specified interventions, which can lead to a potentially infinite number of outcomes. To manage complexity, the language is restricted to explicitly reference countably many values and interventions. However, despite improving on expressivity of SEM-based causal models, it is not clear how one can formalize various aspects of action-theoretic/dynamic frameworks there, e.g. non-persistent change supported by fluents, possible dependency between events, temporal order of event occurrence, etc.

Inspired by the aforementioned work on action-theoretic formalization of actual causation [14, 16], in this paper, we propose a formal account of actual cause in hybrid dynamic domains. Our formalization is set within a recently proposed hybrid variant of the situation calculus [20, 21]. We focus on actual primary cause and study causation relative to primitive fluents exclusively. We show that our formalization has some basic intuitive properties and investigate the conditions under which the primary cause persists.

The paper is organized as follows: in the next section, we introduce our base framework, the situation calculus [22] and the recently proposed hybrid temporal situation calculus [20] along with an example therein. In Section 3, we recap previous work on actual causation in the situation calculus. Then in Section 4, we propose our definition of actual primary cause in the HTSC. We also discuss causation relative to our running example and prove some intuitive properties. Finally, we conclude the paper with some discussion in Section 5.

2. Preliminaries

The Situation Calculus

The situation calculus (SC) is a well-known second-order language for representing and reasoning about dynamic worlds [22, 23]. In the SC, all changes are due to named actions, which are terms in the language. Situations represent a possible world history resulting from performing some actions. The constant S_0 is used to denote the initial situation where no action has been performed yet. The distinguished binary function symbol $do(a, s)$ denotes the successor situation to s resulting from performing the action a. The expression $do([a_1, \cdots, a_n], s)$ represents the situation resulting from executing actions $a_1, \cdots, a_n$, starting with situation s. As usual, a relational/functional fluent representing a property whose value may change from situation to situation takes a situation term as its last argument. There is a special predicate $Poss(a, s)$ used to state that action a is executable in situation s. Also, the special binary predicate $s \sqsubset s'$ represents that s' can be reached from situation s by executing some sequence of actions. $s \sqsubseteq s'$ is an abbreviation of $s \sqsubset s' \vee s = s'$. $s < s'$ is an abbreviation of $s \sqsubset s' \wedge Executable(s')$, where $Executable(s)$ is defined as $\forall a', s'.\ do(a', s') \sqsubseteq s \supset Poss(a', s')$, i.e. every action performed in reaching situation s was possible in the situation in which it occurred. $s \leq s'$ is an abbreviation of $s < s' \vee s = s'$.

In the SC, a dynamic domain is specified using a basic action theory (BAT) $\mathcal{D}$ that includes the following sets of axioms: (i) (first-order or FO) initial state axioms $\mathcal{D}_{S_0}$, which indicate what was true initially; (ii) (FO) action precondition axioms $\mathcal{D}_{ap}$, characterizing $Poss(a, s)$; (iii) (FO) successor-state axioms $\mathcal{D}_{ss}$, indicating precisely when the fluents change; (iv) (FO) unique-names axioms $\mathcal{D}_{una}$ for actions, stating that different action terms represent distinct actions; and (v) (second-order or SO) domain-independent foundational axioms Σ, describing the structure of situations [24]. Although the SC is SO, Reiter [22] showed that for certain type of queries ϕ, $\mathcal{D} \models \phi$ iff $\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \models \mathcal{R}[\phi]$, where $\mathcal{R}$ is a syntactic transformation operator called *regression* and $\mathcal{R}[\phi]$ is a SC formula that compiles dynamic aspects of the theory $\mathcal{D}$ into the query ϕ. Thus reasoning in the SC for a large class of interesting queries can be restricted to entailment checking w.r.t a FO theory [22].

Hybrid Temporal Situation Calculus

The SC only allows discrete changes to fluents as a result of actions. However in the real world, many changes are continuous rather than discrete and happens due to the passage of time. For example, a change in room temperature after adjusting the thermostat happens over time, but not immediately. Reiter's temporal SC [22] can model continuous change. In his framework, each action is given a time argument, but the fluents remained atemporal and won't actually change with time; instead, they attain certain values when these time-stamped actions are performed. One cannot query the value of a continuous fluent at some arbitrary time. For example, in a dropping ball scenario, the ball's current position cannot be determined at a specific moment without referencing a time-stamped action.

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 gives the starting time of the situation s. *time* is specified by an axiom $time(a(\vec{x}, t)) = t$ (included in $\mathcal{D}_{S_0}$) for every action function $a(\vec{x}, t)$ in the domain. *start* is specified by the new foundational axiom $start(do(a, s)) = time(a)$. The starting time of S_0 is not enforced. To outlaw temporal paradoxes, the abbreviation $Executable(s)$ is redefined as below.

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

The hybrid temporal situation calculus (HTSC) [20, 21] takes inspiration from hybrid systems in control theory, which are based on discrete transitions between states that continuously evolve over time. In HTSC, SC (atemporal) fluents are preserved, not to represent continuous change, but rather to provide a context within which the values of temporal fluents can change. For instance, the velocity of a ball, a continuous fluent, changes over time when the ball is in a state of falling. Here, the discrete fluent $Falling(s)$ serves as the context that influences how the continuous functional fluent $velocity(t, s)$ varies with time.

HTSC modifies SC's BAT by including the axioms for $time(a)$ and $start(s)$ as well as the new definition of $Executable(s)$ as discussed above in $\mathcal{D}_{S_0}$ and in Σ, respectively. Moreover, in addition to Reiter's successor-state axioms (SSA) [22], which define how fluents change as a result of named actions, HTSC introduces the following state evolution axioms (SEA) [20], each of which defines how a temporal fluent $f(\vec{x})$'s value changes over time.[1]

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

Here $\Phi(\vec{x}, y, t, s)$ represents $\bigvee_{1 \leq i \leq k}(\gamma_i(\vec{x}, s) \wedge \delta_i(\vec{x}, y, t, s))$, where γ_i is a context and δ_i is a relevant formula to be used to compute the temporal fluent f's value when γ_i holds. Ψ denotes $\bigvee_{1 \leq i \leq k} \gamma_i(\vec{x}, s)$, which represents all the mutually exclusive relevant contexts of a temporal fluent. Thus the above SEA states that the value of a temporal fluent $f(\vec{x})$ 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 unchanged. The formula $\delta_i(\vec{x}, y, t, s)$ implicitly or explicitly defines y using some arbitrary (domain-specific) constraints on the variables and fluents.

A hybrid basic action theory [20] is defined as a collection of axioms $\Sigma \cup \mathcal{D}_{ss} \cup \mathcal{D}_{ap} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{se}$, where $\mathcal{D}_{se}$ is the set of state evolution axioms.

Example. We use a simple nuclear power plant (NPP) as our running example. In this domain, we have the following actions: $rupture(p, t)$, i.e. a pipe in plant p ruptures at time t, $csFailure(p, t)$, i.e. the cooling system of p fails at t, $fixP(p, t)$, i.e. a pipe of p is fixed at t, $fixCS(p, t)$, i.e. the cooling system of p is fixed at t, and $mRadiation(p, t)$, representing monitoring of radiation of p at t. For simplicity, we assume a single pipe per plant.

The fluents in this domain consists of $Ruptured(p, s)$ and $CSFailed(p, s)$, representing plant p has a ruptured pipe in situation s and the cooling system of p has failed in s,

[1] Henceforth, all free variables in a sentence are assumed to be universally quantified at the front.

respectively, as well as the temporal fluent $coreTemp(p, t, s)$, which stands for the core temperature of p at time t in situation s.

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

$$Poss(rupture(p, t), s) \equiv true,$$
$$Poss(fixP(p, t), s) \equiv Ruptured(p, s),$$
$$Poss(csFailure(p, t), s) \equiv \neg CSFailed(p, s),$$
$$Poss(fixCS(p, t), s) \equiv CSFailed(p, s),$$
$$Poss(mRadiation(p, t), s) \equiv true.$$

These are self-explanatory. We also have the following successor-state axioms:

$$Ruptured(p, do(a, s)) \equiv \exists t.\ a = rupture(p, t) \vee (Ruptured(p, s) \wedge \neg\exists t.\ a = fixP(p, t)),$$
$$CSFailed(p, do(a, s)) \equiv \exists t.\ a = csFailure(p, t) \vee (CSFailed(p, s) \wedge \neg\exists t.\ a = fixCS(p, t)).$$

Thus, the pipe of plant p has ruptured after action a happens in situation s, i.e. in $do(a, s)$, iff 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. The SSA for *CSFailed* is similar.

For core temperature, we have the following state evolution axiom (here $\gamma_1(p)$, $\gamma_2(p)$, and $\gamma_3(p)$ denote the contexts $Ruptured(p) \wedge CSFailed(p)$, $Ruptured(p) \wedge \neg CSFailed(p)$, and $\neg Ruptured(p) \wedge CSFailed(p)$, respectively):

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

That is, the value of *coreTemp* of p at time t in situation s is dictated by the formula δ_1 if both p's cooling system has failed and its pipe was ruptured, δ_2 if p's pipe was ruptured but its cooling system is working, δ_3 if p's cooling system has failed but its pipe is intact, and 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 $\Delta_1 = 100, \Delta_2 = 35$, 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 $t - \text{start}(s)$ and specifies a rate of temperature increase, 100, 35, or 55 degrees per second, resp., depending on the context γ_i (we could have used more realistic differential equations just as easily).

We assume that there is at least one NPP P_1 in our domain. We also have the following initial state axioms for P_1: $\neg Ruptured(P_1, S_0)$, $\neg CSFailed(P_1, S_0)$, and $coreTemp(P_1, start(S_0), S_0) = -50$. Finally, the unique-names axioms for the above actions can be defined as usual. Henceforth, we use $\mathcal{D}_{npp}$ to refer to the above axiomatization.

3. Actual Achievement Cause in the SC

Given a history of actions/events (often called a scenario) and an observed effect, *actual causation* involves figuring out which of these actions are responsible for bringing about this effect.[2] When the effect is assumed to be false before the execution of the actions in the scenario and true afterwards, the notion is referred to as *achievement (actual) causation*. Based on Batusov and Soutchanski's original proposal [14], Khan and Lespérance (KL) recently offered a definition of achievement cause in the SC [16]. Both of these frameworks assume that the scenario is a linear sequence of actions, i.e. no concurrent actions are allowed. KL's proposal can deal with epistemic causes and effects; e.g., an agent may analyze the cause of some newly acquired knowledge, and the cause may include some

[2]We use actions and events interchangeably.

knowledge-producing action, e.g. *inform*. They showed that an agent may or may not know all the causes of an effect, and can even know some causes while not being sure about others.

To formalize reasoning about effects, KL [16] introduced the notion of *dynamic formulae*. An effect φ in their framework is thus a dynamic formula.[3] Given an effect φ, the actual causes are defined relative to a *narrative* (variously known as a *scenario* or a *trace*) s. When s is ground, the tuple $\langle \varphi, s \rangle$ is often called a *causal setting* [14]. Also, it is assumed that s is executable, and φ was false before the execution of the actions in s, but became true afterwards, i.e. $\mathcal{D} \models \mathit{Executable}(s) \wedge \neg\varphi[S_0] \wedge \varphi[s]$. Here $\varphi[s]$ denotes the formula obtained from φ by restoring the appropriate situation argument into all fluents in φ (see Def. 2).

Note that since all changes in the SC result from actions, the potential causes of an effect φ are identified with a set of action terms occurring in s. However, since s might include multiple occurrences of the same action, one also needs to identify the situations where these actions were executed. To deal with this, KL 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. KL assumed that the initial situation starts at time-stamp 0 and each action increments the time-stamp by one. Thus, their action theory includes the following axioms:

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

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

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

Definition 1. *Let $\vec{x}$, θ_a, and $\vec{y}$ respectively range over object terms, action terms, and object and action variables. The class of* dynamic formulae *φ is defined inductively using the following grammar:*

$$\varphi ::= P(\vec{x}) \mid \mathit{Poss}(\theta_a) \mid \mathit{After}(\theta_a, \varphi) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists \vec{y}.\ \varphi.$$

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

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

Definition 2.

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

We will now present KL's definition of causes in the SC. The idea behind how causes are computed is as follows. Given an effect φ and scenario s, if some action of the action sequence in s triggers the formula φ to change its truth value from false to true relative to

[3] While KL also study epistemic causation, we restrict our discussion to objective causality only. Also, as usual, we will often suppress the situation argument of φ. $\varphi[s]$ denotes the reintroduction of s in φ; see below for a definition.

$\mathcal{D}$, and if there are no actions in s after it that change the value of φ back to false, then this action is an actual cause of achieving φ in s. Such causes are referred to as *primary* causes:

Definition 3 (Primary Cause [16])**.**

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

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

Now, note that a (primary) cause a might have been non-executable initially. Also, a might have only brought about the effect conditionally and this context condition might have been false initially. Thus earlier actions on the trace that contributed to the preconditions and the context conditions of a cause must be considered as causes as well. The following definition captures both primary and indirect causes.[4]

Definition 4 (Actual Cause [16])**.**

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

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 $(a, ts, [\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)], s')$ is in P (i.e. a executed at ts is a direct or indirect cause of $[\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)]$ in s'), then (a, ts, φ, s) is in P. Here the effect $[\mathit{Poss}(a') \wedge \mathit{After}(a', \varphi)]$ requires a' to be executable and φ to hold after a'.

Example. We will illustrate causation in the SC using a version of our example. Assume a SC BAT for this domain, $\mathcal{D}^{SC}_{npp}$ that only includes the atemporal variants of the last 3 actions, i.e. $\mathit{mRadiation}(p)$, $\mathit{csFailure}(p)$, and $\mathit{fixCS}(p)$, the $\mathit{CSFailed}(p, s)$ fluent, and the associated initial state axioms, action precondition axioms, successor-state axioms, and unique-names axioms. Within this framework, consider the scenario $\sigma_1 = \mathit{do}([\mathit{mRadiation}(P_1), \mathit{csFailure}(P_1), \mathit{fixCS}(P_1), \mathit{mRadiation}(P_1), \mathit{csFailure}(P_1), \mathit{mRadiation}(P_1)], S_0)$ and the observed effect $\varphi_1 = \mathit{CSFailed}(P_1)$, for power-plant P_1. We can show the following.

Proposition 1.

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

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

Proposition 2.

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

[4] In this, we need to quantify over situation-suppressed DF. Thus we must encode such formulae as terms and formalize their relationship to the associated SC formulae. This is tedious but can be done essentially along the lines of [25]. We assume that we have such an encoding and use formulae as terms directly.

Explaining backwards, the second $csFailure(P_1)$ action executed at time-stamp 4 is a cause as it is a direct cause. The $fixCS(P_1)$ action is a cause since had it not for this action, the primary cause would not have been executable (see action precondition axiom for $csFailure()$). Finally, the first cooling system failure action is required for the $fixCS(P_1)$ to be executable. Since we do not deal with secondary causes in this paper, we will not pursue this discussion further and rather refer the interested reader to [16].

4. Primary Cause in Hybrid Dynamic Domains

We next turn our attention to causation in the HTSC. Our precise contribution here is as follows: first, we will give a general definition of causal setting. We then consider the case where the effect is a primitive atemporal fluent as well as the one where it only involves a primitive temporal fluent. We discuss why KL's definition can be adapted for us for the former and give a new definition for the latter in terms of the former. We use our running example to illustrate our definition. We also study some properties, including one that identifies the conditions under which these causes persist. In the next section, we suggest how one might extend our notion of direct cause relative to primitive temporal fluents as effects to cover the cases where the effect is a conjunction or disjunction of two or more primitive temporal effects (i.e. constraints on the values of primitive temporal fluents).

Hybrid Setting. We start by defining a notion of causal setting in the HTSC.

Definition 5. *A hybrid temporal achievement causal setting is a tuple $\langle \mathcal{D}, \sigma, \varphi \rangle$, where $\mathcal{D}$ is a HTSC BAT, $\sigma \neq S_0$ 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) SC formula such that:*

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

In our framework, φ is a situation- and time-suppressed HTSC formula, which is constrained to be consistent. The exact nature of φ is irrelevant for us as we will only deal with primitive atemporal fluents and conditions on the values of primitive temporal fluents (e.g. $coreTemp(P_1) > 1000$) as effects.[5] Note that the above definition requires the effect φ to be false at the beginning and at the end of the initial situation S_0 and 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 point in time within a situation σ, i.e. at any moment in between the start time and the end time of σ, inclusive. To simplify, we assume that the query is posed relative to the starting time of σ (as enforced by Definition 5). One can always add a subsequent dummy action $noOp$ (that has no effect and that is always possible) and query relative to the updated scenario $do(noOp, \sigma)$ if this is not the case.

As discussed below, a hybrid setting does not necessarily guarantee that the causes of the associated (temporal) effect can always be computed as it 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 2. One last point: when the effect is atemporal, the time arguments above are simply ignored and the definition resembles that of a causal setting in the SC.

Atemporal Primitive Fluents as Effects. We 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 HTSC, it

[5] In the following, we 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.

seems to be natural to adopt this instead of KL's time-step. However, one problem with this is that since actions in HTSC are instantaneous, it is possible for multiple actions to have the same time argument, and as such the execution time $time(a)$ of an action a cannot be used to uniquely identify the occurrence of a on the trace. Thus, for this, we will adopt KL's definition of primary cause [16] given in Definition 3 above; we require that φ for our case is some suitable subclass of HTSC formulae.

Temporal Primitive Fluents as Effects. We will give another definition of primary achievement cause relative to a hybrid setting, but now for the case where the effect involves a primitive temporal fluent instead. Note that for discrete effects, the primary cause (which is an action a) brings about the effect discretely and immediately after its execution. For the temporal case, however, the effect might be only realized after a while, and many irrelevant actions might be executed in between. Thus while defining the primary achievement cause, we need to talk about the achievement time t of the effect within the scenario. But, since time is continuous and thus uncountable, it is not trivial to pinpoint t. For instance, one cannot use a sentence such as $\exists t, t'.\ t' < t \wedge \neg\varphi[t', \sigma] \wedge \varphi[t, \sigma]$ to reveal t within some situation σ; since time is continuous, given a time-point t, the notion of an immediate previous time-point t' is not well defined as there will always be another time-point in the interval (t', t), e.g. $(t' + t)/2$. One solution to this problem is to instead consider an interval (t_1, t_2) within the achievement situation over which the effect was achieved. For simplicity, t_1 and t_2 can be assumed to be the starting time of situations (which can be also represented using the time of the associated actions for non-initial situations). In our formalization, we will thus evaluate the effect relative to such t_1 and t_2 to determine the "achievement situation" of the effect, as can be seen in the definition of $AchvSit(\cdot, \cdot, \cdot)$ below.

The intuition behind our definition is as follows. Recall that in HTSC, the values of temporal fluents can change when certain contexts are enabled. Contexts, which are discrete fluents, on the other hand change due to the execution of actions. Thus when determining the primary cause of some temporal fluent having a certain value, we need to identify the last context γ that was enabled when the fluent acquired this value (i.e. the context γ of the achievement situation s_φ), and the action a that caused this context. Since contexts are mutually exclusive, γ must have been the only enabled context in s_φ and a must have been the last action whose contribution brought about the temporal effect under consideration.[6]

In the following, we give the definition of primary cause relative to a hybrid 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 (indexed by i) that are associated with the temporal fluent f (see the state evolution axioms defined above).

Definition 6 (Primary Achievement Cause (Primitive Temporal Case))**.**

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

That is, action a executed at time-stamp ts directly causes the situation- and time-suppressed effect formula φ in scenario s iff the achievement situation of φ in s is s_φ, and a executed in some earlier situation with time-stamp ts directly caused some relevant context γ_i^f for the temporal fluent f in φ in scenario s_φ.

[6] There can be other secondary/indirect causes, but we are only concerned with primary causes here.

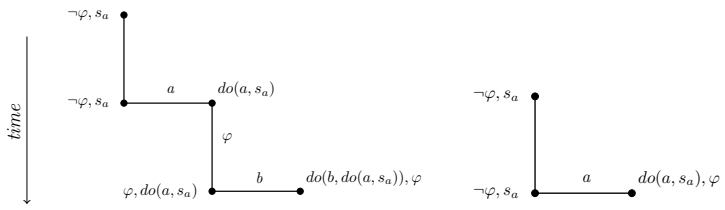

Figure 1. Illustration of the two cases in the definition of achievement situation.

We now give a definition for achievement situations.

$$
\begin{aligned}
&\mathit{AchvSit}(s_\varphi, \varphi, s) \stackrel{\text{def}}{=} \\
&([\exists a, b, s_a.\ S_0 \leq s_a < do(a, s_a) < do(b, do(a, s_a)) \leq s \wedge \neg\varphi[time(a), s_a] \wedge \varphi[time(b), do(a, s_a)] \\
&\quad \wedge (\forall s', s'', t'.\ do(a, s_a) < s' \leq s'' \leq s \wedge start(s') \leq t' \leq start(s'') \supset \varphi[t', s']) \\
&\quad \wedge s_\varphi = do(a, s_a)] \\
&\vee [\exists a, s_a.\ s = do(a, s_a) \wedge \neg\varphi[time(a), s_a] \wedge \varphi[start(do(a, s_a)), do(a, s_a)] \wedge s_\varphi = do(a, s_a)]).
\end{aligned}
$$

The achievement situation of φ in scenario s is s_φ iff either (i) there are two consecutive actions a and b and a situation s_a in the scenario such that φ was false at the end of s_a right before a was executed,[7] but became true by the end of situation $do(a, s_a)$ before the execution of b, and remained true afterwards till the end of the scenario s, in which case s_φ is $do(a, s_a)$; or (ii) there is an action a and some situation s_a such that φ was false at the end of s_a but became true when a happened, i.e. at the start of situation $do(a, s_a)$, which also is the last situation of the trace s, in which case s_φ is $do(a, s_a)$ (which is the same as s). The first disjunct above accommodates cases where there are at least two actions in the scenario, and the effect was false at the end of some situation and became true at some point between the start and the end of the next situation, inclusive. The second one covers the boundary case where φ was instantaneously brought about by the very last action in the scenario. Note that the two cases above are mutually exclusive as the first requires the occurrence of an action *after* the effect φ has been achieved while in the second φ is achieved by the very last action. Figure 1 illustrates these two cases, where horizontal lines represent action execution and vertical lines depict the passage of time within the same situation.

Example. Consider the causal setting $\langle \mathcal{D}_{npp}, \varphi_2, \sigma_2 \rangle$, where $\varphi_2 \stackrel{\text{def}}{=} \mathit{coreTemp}(P_1) > 1000$ and σ_2 is the scenario $do([\mathit{rupture}(P_1, 5), \mathit{csFailure}(P_1, 15), \mathit{mRadiation}(P_1, 20), \mathit{fixP}(P_1, 26)], S_0)$. This is depicted in Figure 2, which also shows the temperature in each situation. Given this, as expected we can show the following result about direct causes.

Proposition 3.

$$\mathcal{D}_{npp} \models \mathit{CausesDirectly}^{prim}_{temp}(\mathit{csFailure}(P_1, 15), 1, \varphi_2, \sigma_2).$$

Note that although $\mathit{csFailure}(P_1, 15)$ is not the last action that happened before the effect φ_2 became true, it is the primary cause. Put otherwise, our definition correctly identified $\mathit{mRadiation}(P_1, 20)$ as one of the irrelevant actions. Indeed $\mathit{csFailure}(P_1, 15)$ is the action that directly caused the context that is active in the achievement situation $S_3 = do([\mathit{rupture}(P_1, 5), \mathit{csFailure}(P_1, 15), \mathit{mRadiation}(P_1, 20)], S_0)$, i.e. γ_1.

[7] Since it is not directly possible to talk about the end time of a situation in HTSC (as the end time does not really exist when the scenario is not known), we will use the start time of the next action to denote this. While in the discussion, we 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.

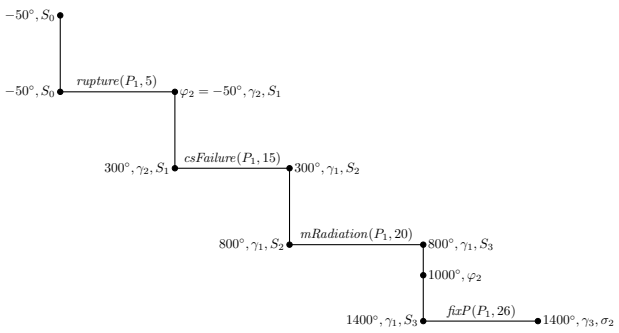

Figure 2. Continuous change in temperature under relevant contexts realized by actions.

Properties. We next discuss a few general properties of our formalization. Let $\mathcal{D}$ include a HTSC BAT and our formalization above.

The first property states that the primary cause of a temporal effect is unique.

Theorem 1 (Uniqueness)**.**

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

This follows directly from Definitions 3 and 6 as by these definitions, both the achievement situation s_φ and the (discrete) direct cause are unique.

Next, we recognise that despite given a proper hybrid setting, primary causes of primitive temporal fluents might not exist (as it may be implicit in the initial situation S_0).

Theorem 2 (Implicit Primary Cause)**.** *Assume that φ is a constraint on the values of a primitive temporal fluent f. Then we have:*

$$\begin{aligned}\mathcal{D} \models (&\mathit{ProperHTSCAchvCausalSetting}(\varphi, \sigma)\\ &\wedge \exists s_\varphi.\ \mathit{AchvSit}(s_\varphi, \varphi, \sigma) \wedge \exists i.\ \gamma_i^f[s_\varphi] \wedge (\forall s'.\ S_0 \leq s' \leq s_\varphi \supset \gamma_i^f[s']))\\ &\supset \neg\exists a, ts.\ \mathit{CausesDirectly}^{prim}_{temp}(a, ts, \varphi, \sigma)),\end{aligned}$$

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 can be proven by showing that since the context γ_i^f active in the achievement situation s_φ was true throughout the interval (S_0, s_φ), the achievement cause of γ_i^f in s_φ simply does not exist (recall that Definition 3 requires γ_i^f to be false before the action that caused it happened).

Finally, we study the conditions under which primary achievement causes persist when the scenario changes.

Theorem 3 (Persistence)**.**

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

That is, if an action a executed in ts is the primary cause of an effect φ in scenario s, then a in ts remains the primary cause of φ in all subsequent situations/scenarios s^* if φ does not change after it was achieved in s. This is because since the achievement situation s_φ does not change in the extended scenario, by Definition 6, neither does the primary cause of φ.

Note that this holds even when the context changes and so does the value of the associated fluent f in φ (as long as φ itself remains unchanged).

5. Conclusion

In this paper, we proposed a formalization of primary achievement cause in hybrid dynamic domains. To the best of our knowledge, ours is the first attempt to deal with causation in hybrid temporal action-theoretic frameworks (the only other formal attempt to this end that we are aware of is the work [19] discussed in the introduction; however, as we mentioned there, that framework is not based on a proper action-theory and thus has many expressive limitations).

Our current proposal is nonetheless limited in many ways. For instance, we only dealt with (conditions on the values of) primitive fluents as effects. Moreover, we did not handle indirect causes. However, our attempt shows that determining causes even under such strong restrictions requires careful modeling and reasoning.

With some effort, our proposal can be extended to compute the primary cause of compound effects, those that are built from disjunctions and conjunctions of primitive temporal effects (i.e. conditions on primitive temporal fluents). For example, if φ is a conjunction of the form $\varphi_1 \wedge \varphi_2$, the action a_1 executed in timestamp ts_1 is the primary achievement cause of φ_1 in σ, and a_2 executed in ts_2 is the primary achievement cause of φ_2 in σ, then one can take the latest between a_1 and a_2 as the primary cause of φ in σ; this is because the last action that contributed to the effect should be considered as the primary cause. The case for disjunctions (i.e. when $\varphi = \varphi_1 \vee \varphi_2$) is a little more trickier and simply taking the latest cause is not adequate as the primary cause of φ will also depend on which of these two disjuncts was actually achieved, as well as on whether both of these disjuncts were achieved at the same time. We leave these for future work. Finally, in the future, we also plan to extend this to discover indirect causes. This should be doable along the same lines as in [14, 16], but perhaps with the help of the newly proposed regression operator in the HTSC [20].

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number RGPIN-2022-03433].

References

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

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

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

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

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

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

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

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

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

[10] H. A. Simon. "Causal Ordering and Identifiability". In: *Models of Discovery. Boston Studies in the Philosophy of Science* 54 (1977).

[11] C. Glymour, D. Danks, B. Glymour, F. Eberhardt, J. D. Ramsey, R. Scheines, P. Spirtes, C. M. Teng, and J. Zhang. "Actual Causation: A Stone Soup Essay". In: *Synthese* 175.2 (2010), pp. 169–192.

[12] F. Leitner-Fischer and S. Leue. "Causality Checking for Complex System Models". In: *Verification, Model Checking, and Abstract Interpretation, 14th International Conference, VMCAI 2013, Rome, Italy, January 20-22, 2013. Proceedings*. Ed. by R. Giacobazzi, J. Berdine, and I. Mastroeni. Vol. 7737. Lecture Notes in Computer Science. Springer, 2013, pp. 248–267.

[13] V. Batusov and M. Soutchanski. "Situation Calculus Semantics for Actual Causality". In: *Proceedings of the Thirteenth International Symposium on Commonsense Reasoning, COMMONSENSE 2017, London, UK, November 6-8, 2017*. Ed. by A. S. Gordon, R. Miller, and G. Turán. Vol. 2052. CEUR Workshop Proceedings. CEUR-WS.org, 2017.

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

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

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

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

[18] V. Yazdanpanah, E. H. Gerding, S. Stein, M. Dastani, C. M. Jonker, T. J. Norman, and S. D. Ramchurn. "Reasoning about responsibility in autonomous systems: challenges and opportunities". In: *AI Soc.* 38.4 (2023), pp. 1453–1464.

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

[20] V. Batusov, G. De Giacomo, and M. Soutchanski. "Hybrid Temporal Situation Calculus". In: *Advances in Artificial Intelligence - 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, Kingston, ON, Canada, May 28-31, 2019, Proceedings*. Ed. by M. Meurs and F. Rudzicz. Vol. 11489. Lecture Notes in Computer Science. Springer, 2019, pp. 173–185.

[21] V. Batusov, G. De Giacomo, and M. Soutchanski. "Hybrid temporal situation calculus". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*. Ed. by C. Hung and G. A. Papadopoulos. ACM, 2019, pp. 1162–1164.

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

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

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

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