Towards A Logical Account of Epistemic Causality

Shakil M. Khan Mikhail Soutchanski

Department of Computer Science
Ryerson University
Toronto, Canada

{shakilmkhan,mes}@scs.ryerson.ca

Reasoning about observed effects and their causes is important in multi-agent contexts. While there has been much work on causality from an objective standpoint, causality from the point of view of some particular agent has received much less attention. In this paper, we address this issue by incorporating an epistemic dimension to an existing formal model of causality. We define what it means for an agent to know the causes of an effect. Then using a counterexample, we prove that epistemic causality is a different notion from its objective counterpart.

1 Introduction

Research on actual causality involves finding in a given narrative (trace) the event that caused an effect. Pearl [25, 26] was a pioneer to lead a computational enquiry in actual causality. The research was later continued by Halpern and Pearl [12, 15] and others [8, 17, 18, 13, 14]. Unfortunately, as argued by Glymour et al. [9], most of these accounts are developed by analyzing a handful of simple examples, and then validated relative to our intuition for these examples, a process which Gößler et al. [11] referred to as TEGAR (i.e. Textbook Example Guided Analysis Refinement). As such, even after multiple revisions, these definitions continue to suffer from various conceptual problems such as the early preemption problem and the over-determination problem. For instance, despite claims to the contrary, the definitions given in [14] suffer from the problem of preemption, which occurs when two competing events try to achieve the same effect and the latter of these fails to do so as the earlier one has already achieved the effect (see [31] and [4] for a discussion).

In an attempt to address these issues, Batusov and Soutchanski [2, 3] recently proposed a new definition of actual causality that is based on a well developed and expressive formalization of actions and change, namely the situation calculus [23, 27]. The definition is derived from first principles and does not follow a TEGAR scheme. One of the advantages of their work is that it allows one to reason about actual causes of *quantified* effects. As argued in [3], their definition (a version of which can be found in Section 3 below) does not suffer from preemption and can handle the more problematic examples well; e.g. both the disjunctive and the conjunctive versions of the well-known "Forest Fire" example [15, 14] are properly handled. In our previous work [20], we showed that this formalization of actual causality has some intuitive properties. In particular, we proved that the computed causes of any given effect and the "causal chain" (relative to a "causal setting", as defined in Section 3 below) are *unique* for any given model, and *sufficient*, but can be *unnecessary* in the sense that removing the relevant events from the trace may still bring about the desired effect. The latter allows for other non-cause events –events that were previously preempted by the some relevant events– to bring about the effect. We also proved that this formalization of causal analysis is modular since causal analysis can be performed by examining only the relevant subset of the system specification. Finally, we discussed how this definition can be utilized for further processing the reconstructed event traces obtained from a discrete event system (DES) based diagnoser.

Submitted to:
CREST 2019

Note that, an important advantage of this framework is that it is based on a well developed formal theory of actions, namely the situation calculus, and as such it automatically inherits many of the advantages of the underlying framework. In this paper, we present our work in progress on extending the notion of actual causality with one such aspect, namely previous work on knowledge within the situation calculus. This allows for a first person's perspective of causality, i.e. causality relativised to the mental states of an agent, specially to that of her knowledge.[1] Equipped with such technical machinery, agents can then reason about the causes of change in each other's mental states. This reasoning ability can be useful in distributed systems, be they systems of interacting agents or networked hosts, where each subsystem/agent has to take individual actions and engage in communication with other subsystems/agents. Finally, we envision using this formalism for developing, among other things, notions such as trust, moral responsibility, and blameworthiness within serious games/multi-agent settings.

The main contribution of this paper is two-fold. First, we incorporate a notion of knowledge within an expressive formal framework for causal analysis. To this end, we define a notion of knowledge of the actual causes of an observed effect, i.e. knowledge relative to a *causal setting* (see below). This unleashes the power of causal analysis by allowing agents to reason about the causes of observed effects, including each other's knowledge (and goals). Secondly, using a simple counter-example, we formally show that, as expected, epistemic causality is a different notion than its objective counterpart in the sense that in different epistemic alternatives, different causes may bring about the same effect even if the narrative remains the same.

While doing this, we also identify a limitation of the formalism proposed in [2] and discuss how one can address this issue. To be specific, the definition in [2] assumes that all actions are fully observable. While we do not solve this issue in this paper, we discuss how this constraint can be relaxed by incorporating belief and belief revision instead of knowledge within this framework.

The paper is organized as follows. In the next section, we outline the situation calculus. In Section 3, we give a version of the definition of actual achievement causes proposed by Batusov and Soutchanski [3]. In Section 4, we review previous work on knowledge in the situation calculus. Then in Section 5, we propose a model of epistemic causality and using an example show how this notion differs from the original notion of causality. Finally, we summarize our results and conclude in Section 6.

2 The Situation Calculus

The situation calculus [23] is a popular formalism for modeling and reasoning about dynamic systems. Here, we use a version as described by Reiter [27]. There are three basic sorts in the language, *situation*, *action*, and a catch-all *object* sort. A situation represents a sequence of actions. A special constant S_0 is used to denote the initial situation where no actions has yet been performed. Here, and subsequently, we use lower-case arguments for variables and upper-case arguments to represent constants. However, function and predicate symbols start with lower-case letters. There is a distinguished binary function symbol *do*, where $do(a,s)$ denotes the successor situation to s resulting from performing the action a. For example, if $drive(agt,i,j)$ stands for an autonomous driving agent agt's action of driving the car from point i to point j, then the situation term $do(drive(Agt_1,I_1,J_1),S_0)$ denotes the situation resulting from Agt_1's driving the car from I_1 to J_1 when the world is in situation S_0. Also, $do(drive(Agt_1,J_1,K_1),do(turn(Agt_1,J_1),do(drive(Agt_1,I_1,J_1),S_0)))$ is a situation denoting the world history consisting of the following sequence of actions: $[drive(Agt_1,I_1,J_1),turn(Agt_1,J_1),$ $drive(Agt_1,J_1,K_1)]$. Thus the situations can be viewed as branches in a tree, where the root of the tree

[1] Handling belief revision complicates the framework somewhat, and therefore we focus on knowledge rather than belief.

is S_0 and the edges represent actions. $do([a_1,\cdots,a_n],s)$ is used to denote the complex situation term obtained by consecutively performing $a_1,\cdots,a_n$ starting from s. Also, the notation $s \sqsubset s'$ means that situation s' can be reached from situation s by executing a sequence of actions. $s \sqsubseteq s'$ is an abbreviation of $s \sqsubset s' \vee s = s'$. Relations whose truth values vary from situation to situation are called relational fluents, and are denoted by predicate symbols taking a situation term as their last argument. There is a special predicate $Poss(a,s)$ used to state that action a is possible in situation s. Finally, a situation s is called *executable* if every action in its history was possible in the situation where it was performed:

$$executable(s) \overset{\text{def}}{=} \forall a',s'.\, do(a',s') \sqsubseteq s \rightarrow Poss(a',s').$$

Following Reiter, we use a basic action theory (**BAT**) $\mathcal{D}$ that includes the following set of axioms: (1) action precondition axioms $\mathcal{D}_{apa}$, one per action a characterizing $Poss(a,s)$, (2) successor-state axioms $\mathcal{D}_{ssa}$, one per fluent, that succinctly encode both effect and frame axioms and specify exactly when the fluent changes, (3) initial state axioms $\mathcal{D}_{S_0}$ describing what is true in S_0, (4) unique name axioms for actions $\mathcal{D}_{una}$, and (5) domain-independent foundational axioms Σ describing the structure of situations.

Example

We use a simple autonomous/driverless car domain as our running example. We have at least one such car/agent, C. An agent c can drive from intersection i to intersection j (and turn at intersection i) by executing the $drive(c,i,j)$ (and $turn(c,i)$, resp.) action.[2] The geometry of the intersections is captured using the non-fluent relation $connected(i,j)$, which states that there is a street from intersection i to j. Unfortunately, due to poor design choices, the agents are vulnerable to over-the-air attacks by hackers. In particular, an agent c's Turn Collision Avoidance System (T-CAS) can be easily corrupted by executing the $hack(c)$ action. If its T-CAS is corrupted, turning a car damages it. Finally, initially all the cars are undamaged and their T-CAS are uncorrupted.

There are three fluents in this domain, $at(c,i,s)$, $corrupted(c,s)$, and $damaged(c,s)$, which mean that the agent c is at location i in situation s, c's T-CAS is corrupted in s, and c is damaged in s, respectively.

We now give the domain-dependent axioms specifying this example domain. First, the preconditions for $drive(c,i,j)$, $turn(c,i)$, and $hack(c)$ can be specified using action precondition axioms (**APA**) as follows (henceforth, all free variables in a sentence are assumed to be universally quantified):

$$\begin{aligned}
&(a).\ Poss(drive(c,i,j),s) \leftrightarrow at(c,i,s) \wedge i \neq j \wedge connected(i,j),\\
&(b).\ Poss(turn(c,i),s) \leftrightarrow at(c,i,s),\\
&(c).\ Poss(hack(c),s).
\end{aligned}$$

That is, (a) an agent c can drive from intersection i to j in some situation s if and only if c is at intersection i in situation s, i and j refer to different intersections, and there is a street connecting i and j; (b) c can turn at intersection i in situation s if and only if c is at i in s; and (c) a hacker can always hack c.

Moreover, the following successor-state axioms (**SSA**) specify how exactly the fluents *at*, *corrupted*, and *damaged* changes value when an action a happens in some situation s:

$$\begin{aligned}
&(d).\ at(c,i,do(a,s)) \leftrightarrow (\exists j(a = drive(c,j,i)) \vee (at(c,i,s) \wedge \neg\exists j(a = drive(c,i,j)))),\\
&(e).\ corrupted(c,do(a,s)) \leftrightarrow (a = hack(c) \vee corrupted(c,s)),\\
&(f).\ damaged(c,do(a,s)) \leftrightarrow ((corrupted(c,s) \wedge \exists i(a = turn(c,i))) \vee damaged(c,s)).
\end{aligned}$$

[2] For brevity, we ignore the turn direction, the traffic light requirements, etc., although we could have easily modeled these.

That is, (d) an agent c is at location i in the situation resulting from executing some action a in situation s (i.e. in $do(a,s)$) if and only if a refers to c's action of driving from location j to i, or she was already at i in s and a is not the action of her driving to another location j; (e) c's T-CAS is corrupted after action a happens in situation s if and only if a is the action of hacking c or her T-CAS was already corrupted in s; and (f) c is damaged after action a happens in situation s if and only if c's T-CAS was corrupted in s and a refers to the action of turning c at some intersection i, or c was already damaged in s.

Furthermore, the following initial state axioms say that initially (g) all the agent's T-CAS are uncorrupted, (h) all agents are undamaged, and (i) they are located at intersection I:

$$(g).\ \forall c(\neg corrupted(c,S_0)), \qquad (h).\ \forall c(\neg damaged(c,S_0)), \qquad (i).\ \forall c(at(c,I,S_0)).$$

We assume for simplicity three intersections connected with two streets:

$$(j).\ \forall i,j.\ connected(i,j) \leftrightarrow ((i=I \wedge j=J) \vee (i=J \wedge j=I) \vee (i=J \wedge j=K) \vee (i=K \wedge j=J)).$$

Also, for simplicity and illustration, we assume the domain closure axiom (k) for the intersections, stating that there are only three intersections I, J, and K in this domain:

$$(k).\ \forall i(i=I \vee i=J \vee i=K).$$

However, we do not require a domain closure axiom for cars/agents, as their number can be unknown. Finally, we need unique names axioms (l), stating that I, J, and K refer to three different intersections:

$$(l).\ I \neq J \wedge I \neq K \wedge J \neq K.$$

Also the following unique names for actions axioms (**UNA**) say that (m) *drive*, *turn*, and *hack* refer to different actions, and (n) two actions with the same function symbol refer to the same action if their arguments are the same (these are necessary for the above successor-state axioms to work properly):

$$(m).\ \forall c_1,c_2,i,j,k(drive(c_1,i,j) \neq turn(c_2,k) \wedge drive(c_1,i,j) \neq hack(c_2) \wedge turn(c_1,k) \neq hack(c_2)),$$

$$(n).\ \forall c_1,c_2,i,j,k,l((drive(c_1,i,j) = drive(c_2,k,l) \rightarrow (c_1 = c_2 \wedge i=k \wedge j=l))$$
$$\wedge (turn(c_1,i) = turn(c_2,j) \rightarrow (c_1 = c_2 \wedge i=j)) \wedge (hack(c_1) = hack(c_2) \rightarrow (c_1 = c_2)).$$

Henceforth, we use $\mathcal{D}_{ac}$ to refer to the above axiomatization of the autonomous car domain.

Regression in the Situation Calculus

BATs employ *regression*, a powerful reasoning mechanism for answering queries about the future. Given a query "does ϕ hold in the situation obtained by performing the ground action α in situation s, i.e. in $do(\alpha,s)$?",[3] the single-step regression operator ρ transforms it into an equivalent query "does ψ hold in situation s?", eliminating action α by compiling it into ψ. The expression $\rho[\phi,\alpha]$ denotes such a logically equivalent query obtained from the formula ϕ by replacing each fluent atom F in ϕ with the right-hand side of the successor-state axiom for F where the action variable a is instantiated with the ground action α, and then simplified using unique name axioms for actions and constants. One can prove that given a BAT $\mathcal{D}$, a formula $\phi(s)$ *uniform in s* (meaning that it has no occurrences of *Poss*, $\sqsubseteq$, other situation terms besides s, and quantifiers over situations), and a ground action term α, we have that $\mathcal{D} \models \forall s.\ \phi(do(\alpha,s)) \leftrightarrow \rho[\phi(s),\alpha]$. One can also obtain a similar regression operator $\mathcal{R}$ by repetitive recursive application of ρ. Reiter [27] showed that for a *regressable* query ϕ, $\mathcal{D} \models \phi$ if and only if $\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \models \mathcal{R}[\phi]$. Regression thus simplifies entailment checking by compiling dynamic aspects of the theory into the query.

[3] A ground term is one whose constituents are ground sub-terms and constants, i.e. that contains no variables.

Example (Continued)

Let us compute $\rho[damaged(C,do(turn(C,K),S^*)),turn(C,K)]$, for some situation S^*. From the right-hand side of the SSA (f) above and by substituting action variable a by $turn(C,K)$, object variables c by C and i by K, and situation variable s by S^*, the result of single-step regression $\rho[damaged(C,do(turn(C,K),S^*)),turn(C,K)]$ amounts to $(corrupted(C,S^*) \wedge turn(C,K) = turn(C,K)) \vee damaged(C,S^*)$. Using the unique names axiom (n) above, the result of ρ can be simplified to $corrupted(C,S^*) \vee damaged(C,S^*)$. Thus, in this example ρ allows us to answer the query $damaged(C,do(turn(C,K),S^*))$ relative to situation $do(turn(C,K),S^*)$ by reducing it to the equivalent simpler query $corrupted(C,S^*) \vee damaged(C,S^*)$ that only mentions the preceding situation S^* and does not mention the situation $do(turn(C,K),S^*)$.

3 Actual Achievement and Maintenance Causes

Given a trace of events,[4] *actual achievement causes* are some of the events that are behind achieving an effect while *actual maintenance causes* are those which are responsible for mitigating the threats to the achieved effect. There can be also cases of subtle interactions of these two. In this section, we review how one can define achievement causality in the situation calculus [3]. An effect in this framework is a situation calculus formula $\phi(s)$ that is uniform in s and that may include quantifiers over object variables. Given an effect $\phi(s)$, the actual causes of ϕ are defined relative to a *causal setting* that includes a BAT $\mathscr{D}$ representing the domain dynamics, and a "narrative" (a trace of events) σ, representing the ground situation, where the effect was observed.

Definition 1 (Causal Setting)**.** *A causal setting is a tuple $\langle \mathscr{D}, \sigma, \phi(s) \rangle$, where $\mathscr{D}$ is a BAT, σ is a ground situation term of the form $do([a_1, \cdots, a_n], S_0)$ with ground action functions $a_1, \cdots, a_n$ such that $\mathscr{D} \models executable(\sigma)$, and $\phi(s)$ is a situation calculus formula uniform in s such that $\mathscr{D} \models \phi(\sigma)$.*

As the theory $\mathscr{D}$ does not change, we will often suppress $\mathscr{D}$ and simply write $\langle \sigma, \phi(s) \rangle$. Also, here we require ϕ to hold by the end of the narrative σ, and thus ignore the cases where ϕ is not achieved by the actions in σ, since if this is the case, the achievement cause truly does not exist.

Note that since all changes in the situation calculus result from actions, we identify the potential causes of an effect ϕ with a set of ground action terms occurring in σ. However, since σ might include multiple occurrences of the same action, we also need to identify the situations when these actions were executed. Now, the notion of the achievement cause of an effect suggests that if some action α of the action sequence in σ triggers the formula $\phi(s)$ to change its truth value from false to true relative to $\mathscr{D}$, and if there are no actions in σ after α that change the value of $\phi(s)$ back to false, then α is the actual cause of achieving $\phi(s)$ in σ.

When used together with the single-step regression operator ρ, the above interpretation of achievement condition not only identifies the single action that brings about the effect of interest, but also captures the actions that build up to it. Intuitively, $\rho[\phi, \alpha]$ specifies the weakest condition that must hold in a previous situation (let us call it σ') in order for ϕ to hold after performing the action α in situation σ', i.e. in situation $do(\alpha, \sigma')$. Thus, if the action α is an achievement cause of ϕ in situation $do(\alpha, \sigma')$, then we can use the single-step regression operator ρ to obtain a formula that holds at situation σ' and constitutes a necessary and sufficient condition for the achievement of $\phi(s)$ via the action α. This new formula may have an achievement cause of its own which, by virtue of the action α, also constructively contributes to the achievement of ϕ. By repeating this process, we can uncover the entire chain of actions that incrementally build up to the achievement of the ultimate effect. At the same time, we must

[4] We do not conceptually distinguish between agents' actions and exogenous/nature's events.

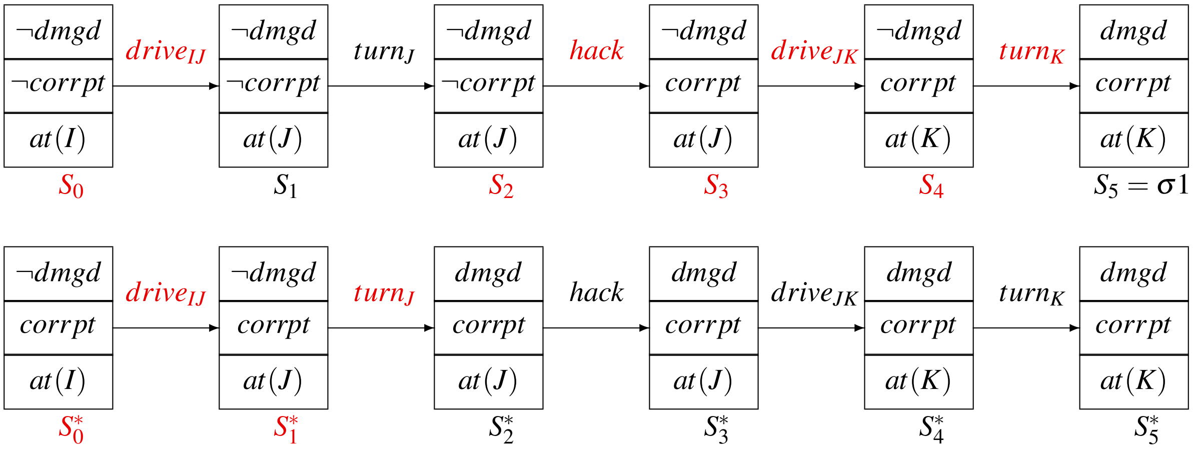

Figure 1: Evolution of fluents relative to narrative σ_1 starting in situations S_0 and S_0^1

not overlook the conditions that make the execution of the action α in situation σ even possible, which are conveniently captured by the right-hand side of the action precondition axiom for α and may have achievement causes of their own.

The following inductive definition formalizes this intuition. Let $\Pi_{apa}(\alpha,\sigma)$ be the right-hand side of the action precondition axiom for action α with the situation term replaced by situation σ.

Definition 2 (Achievement Cause). *A causal setting $\mathscr{C} = \langle\sigma,\phi(s)\rangle$ satisfies the achievement condition of ϕ via the situation term $do(\alpha^*,\sigma^*) \sqsubseteq \sigma$ if and only if there is an action α' and situation σ' such that:*

$$\mathscr{D} \models \neg\phi(\sigma') \wedge \forall s.\, do(\alpha',\sigma') \sqsubseteq s \sqsubseteq \sigma \rightarrow \phi(s),$$

and either $\alpha^* = \alpha'$ and $\sigma^* = \sigma'$, or the causal setting $\langle\sigma',\rho[\phi(s),\alpha'] \wedge \Pi_{apa}(\alpha',\sigma')\rangle$ satisfies the achievement condition via the situation term $do(\alpha^*,\sigma^*)$. Whenever a causal setting $\mathscr{C}$ satisfies the achievement condition via situation $do(\alpha^*,\sigma^*)$, we say that the action α^* executed in situation σ^* is an achievement cause *in causal setting $\mathscr{C}$.*

Since the process of discovering intermediary achievement causes using the single-step regression operator ρ cannot continue beyond S_0, it eventually terminates. Moreover, since the narrative σ is a finite sequence, the achievement causes of $\mathscr{C}$ also form a finite sequence of situation-action pairs, which we call the *achievement causal chain of $\mathscr{C}$*.

As shown in [2], one can also define the concept of *maintenance causes* by appealing to a counterfactual notion of potential threats in the causal setting that can possibly flip the truth value of the effect ϕ to false, and actions in the narrative that mitigated those threats. In general, actual causes can be either achievement causes or maintenance causes and the causal chain can include both. However, to keep things simple, in this paper we focus exclusively on actual achievement causes.

Example (Continued)

Consider the narrative $\sigma_1 = do([drive(C,I,J),turn(C,J),hack(C),drive(C,J,K),turn(C,K)],S_0)$, i.e. the agent C drives the car from intersection I to J, then C turns at intersection J, then C's T-CAS gets hacked, then C drives to intersection K, and finally C turns at K (see the top part of Figure 1). We are interested in computing the actual causes of the effect $\phi_1 = damaged(C,s)$. Then according to Definition 2, the causal setting $\langle \phi_1, \sigma_1 \rangle$ satisfies the achievement condition ϕ_1 via the situation term $do(turn(C,K), S_4)$, where $S_4 = do([drive(C,I,J), turn(C,J), hack(C), drive(C,J,K)], S_0)$, so the action $turn(C,K)$ executed in situation S_4 is a primary achievement cause of $damaged(C,s)$.

Moreover, let us compute $\rho[damaged(C,s), turn(C,K)]$ and $Poss(turn(C,K), S_4)$, starting with the former. As shown in Section 2 above, the result of ρ can be simplified to $corrupted(C,S_4) \vee damaged(C, S_4)$. Let us now consider $Poss(turn(C,K), S_4)$; from the right-hand side of action precondition axiom (b) above and by replacing object variables c with C and i with K and situation variable s by S_4, we have $at(C,K,S_4)$. Computing $\rho[damaged(C,s), turn(C,K)] \wedge Poss(turn(C,K), S_4)$ thus gives rise to a new causal setting $\langle (corrupted(C,s) \vee damaged(C,s)) \wedge at(C,K,s), S_4 \rangle$. As can be seen in the top part of Figure 1, this setting satisfies the achievement condition via the action $drive(C,J,K)$, so $drive(C,J,K)$ executed in $S_3 = do([drive(C,I,J), turn(C,J), hack(C)], S_0)$ is a secondary achievement cause. Notice that, while at a first glance driving the car C from intersection J to K may not seem like an intuitive cause for the damage to the car, it can be argued that it is actually a cause. In particular, for C to be damaged, the $turn(C,K)$ action needs to be executable in situation S_4. By the APA (b) above, this means that C must be at intersection K in S_4, which can only be achieved by executing the $drive(C,J,K)$ action in situation S_3. Thus, given narrative σ_1, $drive(C,J,K)$ indeed indirectly contributes to the car C's damage.

Furthermore, this yields yet another setting:

$$\langle \rho[(corrupted(C,s) \vee damaged(C,s)) \wedge at(C,K,s), drive(C,J,K)] \wedge Poss(drive(C,J,K), S_3), S_3 \rangle.$$

Doing simplifications similar to what we did before, we can arrive at the next setting $\langle (corrupted(C, s) \vee damaged(C,s)) \wedge at(C,J,s), S_3 \rangle$, which meets the achievement condition via the action $hack(C)$ executed in situation $S_2 = do([drive(C,I,J), turn(C,J)], S_0)$.

And again, this yields another setting:

$$\langle \rho[(corrupted(C,s) \vee damaged(C,s)) \wedge at(C,J,s), hack(C)] \wedge Poss(hack(C), S_2), S_2 \rangle,$$

which can be simplified to $\langle at(C,J,s), S_2 \rangle$, and meets the achievement condition via $drive(C,I,J)$ executed in situation S_0, and the analysis terminates. Once again, note that while not obvious, $drive(C,I,J)$ indeed contributes to C's subsequent damage as it makes the preconditions of $drive(C,J,K)$ true, which in turn makes that of $turn(C,K)$ true, whose execution damages the car.

Thus, the causal chain obtained is as follows: $\{(turn(C,K), S_4), (drive(C,J,K), S_3), (hack(C), S_2), (drive(C,I,J), S_0)\}$. Note that, since by Axioms (g), (h), and (i), the initial situation is completely specified, it can be shown that this causal chain is unique, i.e. there are no other causal chains relative to causal setting $\langle \mathcal{D}_{ac}, \phi_1(s), \sigma_1 \rangle$.[5]

Note that, in the above example, not all actions from the trace are included in the causal chain, e.g. $turn(C,J)$. To see another example of this, consider the narrative/trace $\sigma_2 = do(\vec{a}, S_0)$, where:

$$\vec{a} = [drive(C,I,J), turn(C,J), hack(C), hack(C), drive(C,J,K), turn(C,K), turn(C,K), drive(C,K,J)].$$

Consider the causal setting $\langle \phi_1, \sigma_2 \rangle$. We can show that by Definition 2, the second $hack(C)$ action executed in $S_3 = do([drive(C,I,J), turn(C,J), hack(C)], S_0)$ is not a cause for this causal setting, nor part of the causal chain relative to this causal setting, since it was preempted by the first $hack(C)$ action. Also, the last two actions, i.e. $turn(C,K)$ executed in $S_6 = do([hack(C), drive(C,J,K), turn(C,K)], S_3)$ and $drive(C,K,J)$ executed in $S_7 = do(turn(C,K), S_6)$ are irrelevant, since they do not contribute to achieving the effect. In general, there might be several irrelevant actions in between the actions included

[5] As mentioned above, we showed this uniqueness property earlier in [20].

in the causal chain. It is important to realize that our definition can clearly distinguish between irrelevant actions and actions in the causal chain.

We can also handle quantified queries. Consider another example, where we have two agents/cars C_1 and C_2. We want to determine the actual causes of $\phi = \exists c, c'(c \neq c' \land \mathit{damaged}(c,s) \land \mathit{damaged}(c',s))$ after each car is hacked and turned, along with some unnecessary actions, starting in situation S_0, say in the narrative $\sigma_3 = do([\mathit{hack}(C_1), \mathit{turn}(C_1,I), \mathit{hack}(C_1), \mathit{drive}(C_2,I,J), \mathit{turn}(C_1,I), \mathit{hack}(C_2), \mathit{turn}(C_2,J), \mathit{drive}(C_1,I,J)], S_0)$. In this case, a similar analysis as above can be used to show that according to our definition the achievement causal chain for this example is as follows: $[(\mathit{turn}(C_2,J),S_6), (\mathit{hack}(C_2),S_5), (\mathit{drive}(C_2,I,J),S_3), (\mathit{turn}(C_1,I),S_1), (\mathit{hack}(C_1),S_0)]$, where $S_1 = do(\mathit{hack}(C_1),S_0)$, $S_2 = do(\mathit{turn}(C_1,I), S_1)$, etc.

4 Knowledge in the Situation Calculus

We now return to our discussion of actual epistemic achievement causes, i.e. causes of an effect from the perspective of an agent. To deal with this, we allow the domain specifier to model agents' mental states, in particular their knowledge. We start by adapting a simple model of knowledge and knowledge change in the situation calculus in this section. In Section 5, we will then extend this notion to handle "knowing the causes of an effect". This allows an agent to reason about causes of various effects.

Knowledge

Following [24, 28], we model knowledge using a possible worlds account adapted to the situation calculus. To allow for the possibility of incomplete initial knowledge, we can now have multiple initial situations. We use $\mathit{Init}(s)$ to mean that s is an initial situation where no action has happened yet, i.e. $\neg\exists a, s'.\ s = do(a,s')$. The actual initial situation is denoted by S_0. Also, $K(\mathit{agt},s',s)$ is used to denote that in situation s, the agent agt thinks that she could be in situation s'. s' is called a K-alternative situation for agent agt in situation s. Using K, the knowledge of an agent, $\mathit{Know}(\mathit{agt},\phi,s)$, is defined as:[6]

Definition 3 (Knowledge). $\mathit{Know}(\mathit{agt},\phi(\mathit{now}),s) \overset{\text{def}}{=} \forall s'.\ (K(\mathit{agt},s',s) \rightarrow \phi(s'))$.

That is, an agent agt knows that the formula ϕ holds in situation s if ϕ holds in all of agt's K-accessible situations in s. As in [29], who generalized the Know and K notation to handle multiple agents by adding an agent argument to them, we adopt this convention; however we will suppress the agent argument when dealing with single agent domains.

Scherl and Levesque [28] extended Reiter's successor-state axiom approach to model the effects of actions on agents' knowledge, combining ideas from Reiter and Moore. As in [28], we require that initial situations can only be K-related to other initial situations:

$$\forall \mathit{agt}, s, s'(\mathit{Init}(s) \land K(\mathit{agt},s',s) \rightarrow \mathit{Init}(s')).$$

As we will see later, the successor-state axiom for K ensures that in all the situations that are K-accessible from $do(a,s)$, a was the last action performed. This along with the above requirement thus implies that all K-related situations share the same action history. We also constrain K to be reflexive, transitive, and

[6] We will use state formulae within the scope of knowledge. A state formula $\phi(s)$ takes a single situation as argument and is evaluated with respect to that situation. We often use ϕ to denote a formula whose fluents may contain a placeholder constant *now* that stands for the situation in which ϕ must hold. $\phi(s)$ is the formula that results from replacing *now* with s. Where the intended meaning is clear, we sometimes suppress the placeholder.

Euclidean in the initial situation to capture the fact that agents' knowledge is true, and that agents have positive and negative introspection:

$$\begin{aligned}
&\forall agt, s(Init(s) \to K(agt, s, s)),\\
&\forall s(Init(s) \to \forall agt, s_1, s_2(K(agt, s_1, s) \land K(agt, s_2, s_1) \to K(agt, s_2, s))),\\
&\forall s(Init(s) \to \forall agt, s_1, s_2(K(agt, s_1, s) \land K(agt, s_2, s) \to K(agt, s_2, s_1))).
\end{aligned}$$

As shown in [28], these constraints then continue to hold after any sequence of actions since they are preserved by the successor state axiom for K.

Example (Continued)

We want to model an agent's knowledge –both about the world and about the actual achievement causes of effects– in the above autonomous vehicle domain. Assume that the agent initially knows that the car C is undamaged and that C is located at intersection I:

$$(o).\ Know(\neg damaged(C, now), S_0), \qquad (p).\ Know(\forall i.\ at(C, i, now) \leftrightarrow i = I, S_0).$$

Thus $\neg damaged(C) \land at(C, I)$ holds in all of her initial K-accessible worlds/situations. Also assume that the agent does not know anything about the integrity of C's T-CAS:

$$(q).\ \neg Know(corrupted(C, now), S_0) \land \neg Know(\neg corrupted(C, now), S_0).$$

Thus, initially there are at least two possible worlds that are K-related to the initial situation S_0, say S_0 and S_0^* (this is depicted in Figure 1). Each of these worlds assigns a different interpretation to the corruptedness of the car's T-CAS.

Knowledge Change

Scherl and Levesque [28] showed how to capture the changes in knowledge of agents that result from actions in the successor state axiom for K. These include knowledge-producing actions that can be either binary sensing actions or non-binary sensing actions. A binary sensing action is a sensing action that senses the truth-value of an associated proposition; e.g., the binary sensing action $sense_{isCorrupted}(agt)$ could be performed to sense whether the agent/car agt's T-CAS is corrupted or not. On the other hand, non-binary sensing actions refer to sensing actions where the agent senses the value of an associated term; e.g., the hypothetical non-binary sensing action $computePercentageOfDamage(agt)$ could be performed to get the percentage of damage to the agent agt. Following [22], the information provided by a binary sensing action is specified using the predicate $SF(a, s)$, which holds if the action a returns the binary sensing result 1 in situation s. A *guarded sensed fluent axiom* is used to associate an action with the property sensed by this action. For example, one might have a guarded sensed fluent axiom to assert that the action $sense_{isCorrupted}(c)$ tells the agent c whether her T-CAS is corrupted in the situation where it is performed, provided that c is located at the garage:

$$at(c, Garage) \to (SF(sense_{isCorrupted}(c), s) \leftrightarrow corrupted(c, s)).$$

Similarly for non-binary sensing actions, the term $sff(a, s)$ is used to denote the sensing value returned by the action. For example, the following guarded sensed fluent axiom asserts that the action

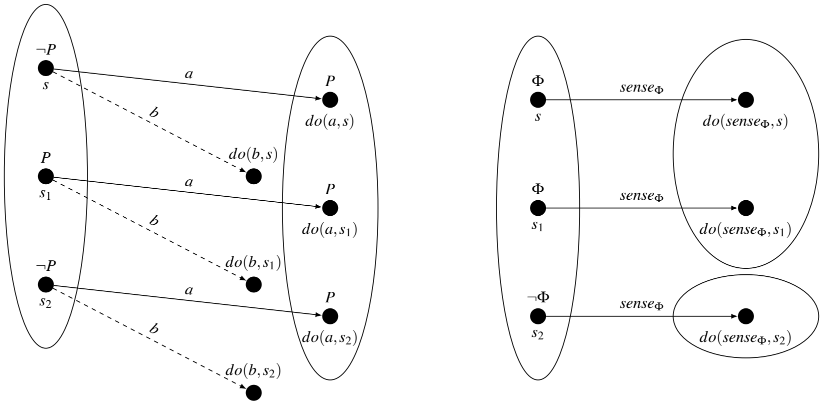

Figure 2: An example of knowledge change

$computePercentageOfDamage(c)$ tells c the percentage of damage to the car, provided that c is at the garage:

$$at(c,Garage) \rightarrow (sff(computePercentageOfDamage(c),s) = percentageOfDamageOn(c,s)).$$

The successor-state axiom for K is defined as follows:[7]

Axiom 1 (Successor-State Axiom for K).

$$\begin{aligned}
&K(agt,s^*,do(a,s)) \leftrightarrow \\
&\qquad \exists s'.[K(agt,s',s) \wedge s^* = do(a,s') \wedge Poss(a,s') \\
&\qquad\quad \wedge ((BinarySensingAction(a) \wedge Agent(a) = agt) \rightarrow (SF(a,s') \leftrightarrow SF(a,s))) \\
&\qquad\quad \wedge ((NonBinarySensingAction(a) \wedge Agent(a) = agt) \rightarrow (sff(a,s') = sff(a,s)))].
\end{aligned}$$

This says that after an action happens, every agent learns that it has happened. Thus, an agent's knowledge is affected by every action in the sense that she comes to know that the action was performed. It is assumed that agents know the successor-state axioms for actions, so the agents also acquire knowledge about the effects of these actions.[8] Moreover, if the action is a sensing action, the agent performing it acquires knowledge of the associated proposition or term. Note that this axiom only handles knowledge expansion, not revision.

We illustrate the successor-state axiom for K using the scenario in Figure 2. In this figure, situations are nodes in the graph, and the edges are labeled by actions. Part of the K-relation is represented by the ovals around the nodes. If a situation s appears in the same oval as another situation s', then $K(agt,s',s)$. Finally, in this figure s denotes the actual situation, i.e. the one representing the true state of the world. First, consider the case for knowledge expansion due to regular actions, as depicted in the left-hand side of Figure 2. Assume that initially s, s_1, and s_2 are K-accessible from each other. Then after action a happens in situation s, according to the successor-state axiom for K, only $do(a,s)$, $do(a,s_1)$, and $do(a,s_2)$ will be accessible from $do(a,s)$, but not $do(b,s_2)$, etc. Thus, in $do(a,s)$ the agent knows that

[7] Lespérance [21] and later others [29] have extended the successor-state axiom for K to support different types of communication actions, but for simplicity we do not consider communication actions here.

[8] One consequence of this is that agents are assumed to be aware of all actions that may happen in the environment. This in part allows us to avoid belief revision and its difficulties.

the action a has just happened and knows that its effects hold. If a makes some property P become true in all K-accessible situations, then the agent knows that P holds afterwards. Next, consider the case for knowledge expansion as a result of knowledge producing actions, as illustrated in the right-hand side of Figure 2. Assume that initially s, s_1, and s_2 are in the same equivalence class wrt K, and that $\Phi(s)$, $\Phi(s_1)$, and $\neg\Phi(s_2)$ holds. Then after the agent senses the value of Φ in s, according to the successor-state axiom for K, only $do(sense_{\Phi}, s)$ and $do(sense_{\Phi}, s_1)$ will be K-accessible from $do(sense_{\Phi}, s)$, but not $do(sense_{\Phi}, s_2)$. Since Φ holds in all situations that are K-accessible from $do(sense_{\Phi}, s)$, the agent will thus know that Φ in $do(sense_{\Phi}, s)$.

5 Actual Epistemic Achievement Causes

We next formalize a notion of knowledge relative to a causal setting. While obvious, we would like to remind the reader that given a causal setting $\mathscr{C} = \langle \mathscr{D}, do([\alpha_1, \cdots, \alpha_n], \sigma), \phi(s)\rangle$ and a causal chain $\mathscr{K} = \{(a_1, s_1) \cdots, (a_m, s_m)\}$ of $\mathscr{C}$, the actions $a_1, \cdots, a_m$ in $\mathscr{K}$ must come from the trace $\alpha_1, \cdots, \alpha_n$. We start by defining the following concept of K-related causal chains.

Definition 4. *Consider two causal settings $\mathscr{C}_1 = \langle \mathscr{D}, do([\alpha_1, \cdots, \alpha_n], \sigma_1), \phi(s)\rangle$ and $\mathscr{C}_2 = \langle \mathscr{D}, do([\alpha_1, \cdots, \alpha_n], \sigma_2), \phi(s)\rangle$ that differ only in the situations where their narratives start, i.e. in initial situations σ_1 and σ_2. Assume that $\mathscr{K}_1$ is a (non-empty) achievement causal chain of causal setting $\mathscr{C}_1$, and $\mathscr{K}_2$ that of $\mathscr{C}_2$. We say that $\mathscr{K}_1$ and $\mathscr{K}_2$ are K-related with respect to the achievement of $\phi(s)$ and action sequence $[\alpha_1, \cdots, \alpha_n]$ if and only if $\mathscr{K}_1$ is of the form $\{(a_1, s_1^1), \cdots, (a_m, s_m^1)\}$ and $\mathscr{K}_2$ is of the form $\{(a_1, s_1^2), \cdots, (a_m, s_m^2)\}$ for some $m > 0$, and for all $1 \le i \le m$, it follows that $\mathscr{D} \models K(agt, s_i^1, s_i^2)$.*

Thus, two causal chains are K-related if they have the same cardinality (i.e. equal number of (action,situation) pairs), and for every (action, situation) pairs in these causal chains, they only (possibly) differ in the situation term, which are required to be K-related. Note that since K is reflexive, this holds trivially when $\mathscr{K}_1 = \mathscr{K}_2$. Intuitively, if two causal chains $\mathscr{K}_1$ and $\mathscr{K}_2$ are K-related, then as far as the agent is concerned, there is no difference between these two causal chains relative to the achievement of the effect ϕ via the execution of the sequence of events $[\alpha_1, \cdots, \alpha_n]$.

Using this, we define the knowledge of a causal chain relative to a causal setting as follows:

Definition 5. *Given a causal setting $\mathscr{C} = \langle \mathscr{D}, \sigma, \phi(s)\rangle$, where $\sigma = do([\alpha_1, \cdots, \alpha_n], s^*)$ for some initial situation s^* and finite $n > 0$, an agent knows in situation σ that $\mathscr{K}$ is the achievement causal chain of $\mathscr{C}$ if and only if:*

- *$\mathscr{K}$ is an achievement causal chain of $\mathscr{C}$, and*
- *for all σ^* such that $\mathscr{D} \models K(\sigma^*, \sigma)$, if $\mathscr{K}^*$ is an achievement causal chain of causal setting $\langle \mathscr{D}, \sigma^*, \phi(s)\rangle$, then causal chains $\mathscr{K}$ and $\mathscr{K}^*$ are K-related relative to the achievement of $\phi(s)$ and the action sequence $[\alpha_1, \cdots, \alpha_n]$.*

That is, an agent knows in situation σ that $\mathscr{K}$ is the achievement causal chain of $\mathscr{C}$ if and only if each of the causal chains computed in the worlds that the agent considers possible in σ is K-related wrt the effect and the trace in $\mathscr{C}$. In the following, we will use the term *actual narrative* to refer to any ground situation $\sigma = do([\alpha_1, \cdots, \alpha_n], s)$ for some $n > 0$, where s is the actual initial situation, i.e. $s = S_0$.

Note that, Definition 5 implicitly specifies that if an agent knows in situation σ that $\mathscr{K}$ is the achievement causal chain of the causal setting $\mathscr{C}$, then the causal chain of setting $\mathscr{C}$ is unique. Since K is reflexive, any causal chain relative to setting $\mathscr{C}$ must be K-related wrt the achievement of ϕ and trace $[\alpha_1, \cdots, \alpha_n]$. Since the setting $\mathscr{C}$ (and as such the situation σ) does not change, this implies the uniqueness of $\mathscr{K}$. By the same token, when an agent has the knowledge of a causal chain relative to a causal

$$\begin{tikzcd}
S_0 \arrow[loop above, "K"] \arrow[r, "drive(C,I,J)"] \arrow[d, "K"] & S_1 \arrow[loop above, "K"] \arrow[r, "turn(C,J)"] \arrow[d, "K"] & S_2 \arrow[loop above, "K"] \arrow[r, "hack(C)"] \arrow[d, "K"] & S_3 \arrow[loop above, "K"] \arrow[r, "drive(C,J,K)"] \arrow[d, "K"] & S_4 \arrow[loop above, "K"] \arrow[r, "turn(C,K)"] \arrow[d, "K"] & \sigma_1 \arrow[loop above, "K"] \arrow[d, "K"] \\
S_0^* \arrow[r, "drive(C,I,J)"] & S_1^* \arrow[r, "turn(C,J)"] & S_2^* \arrow[r, "hack(C)"] & S_3^* \arrow[r, "drive(C,J,K)"] & S_4^* \arrow[r, "turn(C,K)"] & S_5^*
\end{tikzcd}$$

Figure 3: Evolution of the (partial) K relation of the autonomous vehicle agent wrt the actions in σ_1

setting in some situation σ, the causal chain obtained in each of the K-alternative situations to σ (relative to the respective causal settings) is also unique.

Thus, the above definition specifies the conditions under which an agent can be said to know the actual causes of an observed effect. Note that, according to our definition of epistemic causality, it is possible for an agent to not know the causes of a known effect. For example, in decentralized voting protocols, a system/agent that cast a decisive majority is the actual cause of the decision, but another system/agent may not know this since each ballot was secretly cast. In such cases, reasoning must be performed on different epistemic alternatives separately using the original notion of causality.

To appreciate the power of our formalization, note that equipped with the ability to deal with knowledge of the causes of an effect, agents specified in our framework can now reason about the causes of effects relative to/conditioned on what they know. Also, since agents are introspective relative to their knowledge (i.e. they know what they know and know what they don't know), they can also reason about the causes of epistemic effects (changes in their knowledge). Furthermore, incorporating other intentional attitudes within this framework, such as goals and intentions –as was done in [19]– allows agents to reason about the causes of change in their motivations (agents are also introspective relative to their motivational attitudes; see [19] for details). Finally, when multiple agents are involved, this allows agents to reason about each other's knowledge and goals. Therefore, this simple extension unleashes the power of causal analysis and allows agents to reason about the causes of various intentional attitudes.

Example (Continued)

Let $\mathscr{D}_{ac}^{K}$ refer to $\mathscr{D}_{ac}$ along with the above axiomatization of knowledge, knowledge change, and our agent's (initial) knowledge about the autonomous vehicle domain (i.e. Axioms $(o)-(q)$). Using Definitions 4 and 5, we can show the following result on epistemic causality in our autonomous vehicle domain:

Theorem 1. *Given causal setting $\mathscr{C}_{ac} = \langle \mathscr{D}_{ac}, \sigma_1, \phi_1(s) \rangle$ and its achievement causal chain $\mathscr{K}_{ac1} = \{(turn(C,K), S_4), (drive(C,J,K), S_3), (hack(C), S_2), (drive(C,I,J), S_0)\}$. $\mathscr{D}_{ac}^{K}$ entails that the agent does not know in σ_1 that $\mathscr{K}_{ac1}$ is the achievement causal chain of causal setting $\mathscr{C}_{ac}$.*

Proof. By Definition 5, to prove this we need to show that there exists a situation σ^* that is K-accessible from σ_1, i.e. $K(\sigma^*, \sigma_1)$, and an achievement causal chain $\mathscr{K}^*$ of the causal setting $\langle \mathscr{D}_{ac}, \sigma^*, \phi_1(s) \rangle$ is not K-related to the achievement causal chain $\mathscr{K}_{ac1}$ (according to Definition 4). Let us consider an initial situation where the agent C's T-CAS is corrupted, C is undamaged, C is located at intersection I; let us call this situation S_0^*:

$$corrupted(C, S_0^*) \wedge \neg damaged(C, S_0^*) \wedge at(C, I, S_0^*). \tag{1}$$

We claim that the successor to this situation after the actions in the situation σ_1 has happened, i.e. situation $S_5^* = do([drive(C,I,J), turn(C,J), hack(C), drive(C,J,K), turn(C,K)], S_1^*)$, as can be seen in Figure 3, is indeed such a situation σ^*. To show this, we have to show that $K(S_5^*, \sigma_1)$ and that $\mathscr{K}_{ac1}$ and $\mathscr{K}^*$ are not K-related wrt the achievement of $\phi_1(s)$ and the action sequence in σ_1.

We start by showing the former (see Figure 3). Note that it follows from $\mathscr{D}_{ac}^K$ and (1) that S_0^* is K-accessible from the actual initial situation S_0, i.e. $\mathscr{D}_{ac}^K \models K(S_0^*, S_0)$. Moreover, it can be shown that $\mathscr{D}_{ac}^K$ entails that all the actions in σ_1 are known to be executable starting in S_0. Furthermore, since all the actions performed in σ_1 and S_5^* are exactly the same, and since none of these actions are knowledge-producing/sensing actions, by the successor-state axiom for K, it follows that S_5^* is retained in the K-relation in σ_1. Thus we have $K(S_5^*, \sigma_1)$.

Now, computing the achievement causal chain for causal setting $\langle \mathscr{D}_{ac}, S_5^*, \phi_1 \rangle$ using Definition 2 yields the causal chain $\mathscr{K}^* = \{(turn(C,J), S_1^*), (drive(C,I,J), S_0^*)\}$, as can be seen in the bottom part of Figure 1. By Definition 4, $\mathscr{K}_{ac1}$ and $\mathscr{K}^*$ are clearly not K-related wrt the achievement of $\phi_1(s)$ and action sequence $[drive(C,I,J), turn(C,J), hack(C), drive(C,J,K), turn(C,K)]$. □

The above theorem demonstrates that it is possible for the same effect brought about by the same sequence of actions to have different causes in different epistemic alternatives, as is expected. Put otherwise, when mental attitudes are concerned, theories of causation at different levels of (epistemic) nestings need not be related. In the words of Williamson [32], "To say that causal connection is mental does *not* imply that causality is subjective (in the logical sense)".

6 Discussion

The above notion of (objective) causality [3] has motivation that is similar to [10, 30], who also discuss causal analysis relative to traces. However, [10, 30] work with less expressive languages, and unlike us they provide a counterfactual definition of causality. As mentioned earlier and shown in [3], our definition above can correctly compute actual causes even for the more problematic examples with early preemption and overdetermination that create serious difficulties for the structural equations-based approach developed in [25, 26, 15, 14].

Based on this formal notion of causality, in this paper we proposed an account of epistemic causality within a formal theory of action. Our account allows agents to have incomplete initial knowledge. For instance, in our running example, initially the agent doesn't know whether the car's T-CAS system is corrupted or not. We defined what it means for an agent to know the causes of a known effect. We also showed that epistemic causality is a different notion from causality in the sense that given a trace, it is possible to have different causes of the same effect in different epistemic alternatives. Thus, as expected, the agent may or may not know the causes of an effect.

Recently, there has been some work that formalizes causality in an epistemic context. For example, while defining responsibility/blame in legal cases, Chockler et al. [7] modeled an agent's uncertainty of the causal setting using an "epistemic state", which is a pair (K, Pr), where K is a set of causal settings and Pr is a probability distribution over K. Their model is based on structural equations. We on the other hand define epistemic causality based on the more expressive formalism proposed by Batusov and Soutchanski [3]. Moreover, unlike [7], our account incorporates a formal model of domain dynamics and knowledge change. This allows for interesting interplay between causality and knowledge. For instance, in our framework it is possible to specify a domain where the agent does not know the causes of an effect in some situation, but learns them after performing some sensing action. Some of our future work include analyzing such examples as well as defining responsibility and blame using our formalization.

Here, we focus on knowledge and do not deal with belief. Traditionally, agents' knowledge is required to be true while agents are allowed to have incorrect beliefs [16]. Incorporating beliefs yields a more expressive framework, one that allows for causality relative to partially observable actions (and traces). Put otherwise, the agent can now consider different actions in different doxastic alternatives/belief-accessible worlds: given some situation s in the actual narrative, as far as the agent is concerned, the action that was executed in situation s can be any of those performed in one of her belief-accessible worlds (cf. Footnote 6, where we required the agent to know the action that happened in s). The agent can reason about actual causality (under some similar conditions specified in Definition 5), even if she does not know the exact sequence of actions that has been performed since the initial situation. Also, more interesting interplay between objective and epistemic causality can now arise. For instance, an agent may perceive her action to be a cause for some effect ϕ, but in reality it was not, since ϕ was over-determined due to her incorrect beliefs about the world.[9] Similarly, an agent may think her action was not a cause, but in reality it was. While we think that much of our formalization can be extended to deal with beliefs, this requires handling belief revision, which complicates the framework further. We leave this for future.

Finally, we focus in this paper on deterministic actions only. However, there are several proposals on how one can reason about stochastic actions in the situation calculus, e.g. [1, 6, 5]. Dealing with stochastic actions is future work.

Acknowledgements

We thank Yves Lespérance and the anonymous reviewers for useful comments on an earlier version. This work was supported in part by the National Science and Engineering Research Council of Canada and by the Faculty of Science at Ryerson University.

References

[1] Fahiem Bacchus, Joseph Y. Halpern & Hector J. Levesque (1999): *Reasoning about Noisy Sensors and Effectors in the Situation Calculus*. Artificial Intelligence 111(1–2), pp. 171–208.

[2] Vitaliy Batusov & Mikhail Soutchanski (2017): *Situation Calculus Semantics for Actual Causality*. In: Proceedings of the 13th Intl. Symposium on Commonsense Reasoning, COMMONSENSE.

[3] Vitaliy Batusov & Mikhail Soutchanski (2018): *Situation Calculus Semantics for Actual Causality*. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, pp. 1744–1752.

[4] Sander Beckers & Joost Vennekens (2018): *A Principled Approach to Defining Actual Causation*. Synthese 195(2), pp. 835–862.

[5] Vaishak Belle & Hector J. Levesque (2018): *Reasoning about Discrete and Continuous Noisy Sensors and Effectors in Dynamical Systems*. Artificial Intelligence 262, pp. 189–221.

[6] Craig Boutilier, Raymond Reiter & Bob Price (2001): *Symbolic Dynamic Programming for First-Order MDPs*. In: Proceedings of the Seventeenth Intl. Joint Conference on Artificial Intelligence, IJCAI, pp. 690–700.

[7] Hana Chockler, Norman E. Fenton, Jeroen Keppens & David A. Lagnado (2015): *Causal Analysis for Attributing Responsibility in Legal Cases*. In: Proceedings of the 15th Intl. Conference on Artificial Intelligence and Law, ICAIL, pp. 33–42.

[8] Thomas Eiter & Thomas Lukasiewicz (2002): *Complexity Results for Structure-Based Causality*. Artificial Intelligence 142(1), pp. 53–89.

[9] Recall that unlike knowledge, agents' beliefs are not required to be true.

[9] Clark Glymour, David Danks, Bruce Glymour, Frederick Eberhardt, Joseph Ramsey, Richard Scheines, Peter Spirtes, Choh Man Teng & Jiji Zhang (2010): *Actual Causation: A Stone Soup Essay*. *Synthese* 175(2), pp. 169–192.

[10] Gregor Gößler & Daniel Le Métayer (2015): *A General Framework for Blaming in Component-Based Systems*. *Science of Computer Programming* 113, Part 3.

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

[12] Joseph Y. Halpern (2000): *Axiomatizing Causal Reasoning*. *J. Artificial Intelligence Research* 12, pp. 317–337.

[13] Joseph Y. Halpern (2015): *A Modification of the Halpern-Pearl Definition of Causality*. In: *Proceedings of the Twenty-Fourth Intl. Joint Conference on Artificial Intelligence, IJCAI*, pp. 3022–3033.

[14] Joseph Y. Halpern (2016): *Actual Causality*. The MIT Press.

[15] Joseph Y. Halpern & Judea Pearl (2005): *Causes and Explanations: A Structural-Model Approach. Part I: Causes*. *The British Journal for the Philosophy of Science* 56(4), pp. 843–887.

[16] Jaakko Hintikka (1962): *Knowledge and Belief*. Cornell University Press, Ithaca, NY, USA.

[17] Mark Hopkins (2005): *The Actual Cause: From Intuition to Automation*. Ph.D. thesis, Univ. of California L.A.

[18] Mark Hopkins & Judea Pearl (2007): *Causality and Counterfactuals in the Situation Calculus*. *J. Log. Comput.* 17(5), pp. 939–953.

[19] Shakil M. Khan & Yves Lespérance (2010): *A Logical Framework for Prioritized Goal Change*. In: *9th Intl. Conference on Autonomous Agents and Multiagent Systems (AAMAS), Volume 1-3*, pp. 283–290.

[20] Shakil M. Khan & Mikhail Soutchanski (2018): *Diagnosis as Computing Causal Chains from Event Traces*. In: *Proceedings of the AAAI Fall Symposium: Integrating Planning, Diagnosis, and Causal Reasoning (SIP-18)*, AAAI Press.

[21] Yves Lespérance (2003): *On the Epistemic Feasibility of Plans in Multiagent Systems Specifications*. *Logic Journal of the IGPL* 11(2), pp. 161–178.

[22] Hector J. Levesque (1996): *What Is Planning in the Presence of Sensing?* In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI, IAAI, Volume 2*, pp. 1139–1146.

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

[24] Robert C. Moore (1985): *A Formal Theory of Knowledge and Action*. In: *Formal Theories of the Commonsense World*, Ablex, pp. 319–358.

[25] Judea Pearl (1998): *On the Definition of Actual Cause*. Technical Report, Univ. of California L.A.

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

[27] Raymond Reiter (2001): *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press.

[28] Richard B. Scherl & Hector J. Levesque (2003): *Knowledge, Action, and the Frame Problem*. *Artificial Intelligence* 144(1-2), pp. 1–39.

[29] Steven Shapiro (2005): *Specifying and Verifying Multiagent Systems Using CASL*. Ph.D. thesis, Dept. of Computer Science, Univ. of Toronto, Toronto, Ontario, Canada.

[30] Shaohui Wang, Yoann Geoffroy, Gregor Gößler, Oleg Sokolsky & Insup Lee (2015): *A Hybrid Approach to Causality Analysis*. In: *RV 2015 – 6th Intl. Conference on Runtime Verification, LNCS* 9333.

[31] Brad Weslake (2015): *A Partial Theory of Actual Causation*. *British Journal for the Philosophy of Science*.

[32] Jon Williamson (2006): *Dispositional versus Epistemic Causality*. *Minds and Machines* 16(3), pp. 259–276.