Event-Based and Scenario-Based Causality for Computational Ethics

Fiona Berreby, Gauvain Bourgne, Jean-Gabriel Ganascia

Event-Based and Scenario-Based Causality for Computational Ethics

Fiona Berreby
Sorbonne Université - CNRS
UMR 7606, LIP6. Paris, France
fiona.berreby@lip6.fr

Gauvain Bourgne
Sorbonne Université - CNRS
UMR 7606, LIP6. Paris, France
gauvain.bourgne@lip6.fr

Jean-Gabriel Ganascia
Sorbonne Université - CNRS
UMR 7606, LIP6. Paris, France
jean-gabriel.ganascia@lip6.fr

ABSTRACT

This paper makes use of high-level action languages to investigate aspects of causality that are central to ethical reasoning. We identify properties that causal relations assume and that determine how, as well as to what extent, we may ascribe ethical responsibility on their basis. The paper is structured in three parts. First, we present an extension of the Event Calculus that enables the agent to generate plans of actions, with the particularity that they integrate both actions and omissions. Second, we present an account of *event-based* causality that is grounded in the architecture of event preconditions and effects, and that distinguishes four types of causal relations contingent on the nature of the entities that compose them. Namely, it discriminates actions and omissions from automatic events, and produced outcomes from avoided ones. Third, we examine notions of *scenario-based* causality whose role it is to scrutinise and buttress the causal relations previously identified. Inquiring into the other possible versions of modelled scenarios, we account for simple counter-factual validity ("Had I not acted so, would this outcome still be true?"), criticality ("Could anything else have led to this outcome?"), extrinsic necessity ("Had I not produced it, was this outcome even avoidable?"), and elicited necessity ("Have I made this outcome unavoidable?"). The model is implemented in Answer Set Programming.

KEYWORDS

Computational Ethics; Answer Set Programming; Event Calculus; Reasoning about Actions and Change; Causality; Counter-factuals

ACM Reference Format:
Fiona Berreby, Gauvain Bourgne, and Jean-Gabriel Ganascia. 2018. Event-Based and Scenario-Based Causality for Computational Ethics. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

The increasing autonomy and ubiquity of artificial agents urges us to address their capacity to process ethical restrictions and correctly attribute responsibility, be it for their own actions or those of others. Fields as varied as health-care or transportation pose ethical issues that the computational study of ethics has begun to address, as reviewed in [1]. Causality is a concept that is essential to ethical reasoning, as it is the basis for any assignment of responsibility. No blame or praise may be assigned without some account of causal relationship between an agent and an outcome [4]. Our aim is to provide a systematic and adaptable model of causality that can help calibrate such responsibility attribution. It is implemented in Answer Set Programming (for a description, see [23]), and is proposed as a supporting model of the modular framework for ethical reasoning presented in [6].

There are three parts to the paper. To begin, we present in Sect. 2 an extension of the *action model* from [6]. In the initial version, this model based on a modified version of the Event Calculus enabled agents to appraise their environment and choose one action to perform within it. The extension enables the formulation of plans of actions, and, because agents may be held responsible for things they failed to produce or avert, it enables the explicit integration of omissions within these plans. Next, in order to distinctly handle all types of events that can occur within a domain, we define four types of causal relations, accounting for their *direction*, supporting or opposing, and their *strength*, strong or weak. Supporting relations pertain to produced outcomes; opposing relations pertain to avoided ones. Strong relations pertain to the occurrence of automatic events; weak relations to the occurrence of actions and omissions. These relations concern *event-based* causality and constitute Sect. 3. In Sect. 4, we define properties pertaining to *scenario-based* causality, that is, emanating from the exploration of alternative versions of an original scenario. Specifically, we test the capacity of a causal relation between two events to withstand counter-factual and conditional inquiry. Indeed, when we say that an agent has produced a particular outcome, and we aim to ascribe responsibility on that basis, it might be compelling or necessary to know such things as whether the outcome could have been avoided at all, or been produced by other means. We investigate and model four properties of the sort. Related works and future directions of research are respectively discussed in Sects. 5 and 6.

Formally, we chose the use of non-monotonic logic as it allows the manipulation of defeasible generalisations that pervade much common sense reasoning and that are poorly captured by classical logic systems [17]. Inferences where no conclusions are drawn definitely but stay open to modification in the light of further information are prevalent in reasoning about ethical responsibility: such things as the presence of alternative options or extenuating circumstances can overthrow ethical judgement.

2 ACTIONS, OMISSIONS AND PLANS

An agent, broadly, is an entity with the power to act; the demonstration of this capacity is what makes agents liable to blame or praise, both in ethics and in the law. Yet this capacity is not just a matter of performed actions: surely we are to blame if we choose not to rescue a drowning child. Responsibility therefore also pertains to the power to *not* act. Whether there is a fundamental moral

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

difference between actions and omissions is an important point of debate within moral philosophy (eg.[5][19][11]). Answering it affirmatively might for example challenge consequentialism by stating that two identical outcomes should be differently appraised depending on whether an action or omission led to them. As such, it is critical to be able to model them both separately. Whether omissions should be considered special kinds of actions or events, or whether they should be made explicit within causal chains at all are also points of philosophical debate. Our purpose here is not to lend weight to any such accounts, however, in order to reason over them computationally, we have committed to the idea that omissions are a subclass of events. In addition, in order to avoid speaking of omissions as *negative events*, which are philosophically problematic [38][37], we index every omission to an action so that we may speak *in a negative way of positive events*. In everyday life, when we think of someone's failure to act, we imagine the very many ways in which the person might have acted. Our model, rather than interpreting this failure as a single entity and a single omission, would consider this situation to be one in which very many actions could have been performed, and therefore one in which very many omissions occur. Throughout, we refer to actions and omissions as *volitions* - i.e. a decision made.

2.1 Modelling Actions

An action model based on the Event Calculus. In order to model the effects of actions in a domain, we appeal to the action model presented in [6]. It corresponds to the full Event Calculus described in [33], with a number of additions. We introduce non-inertial fluents [27], and a distinction is made between automatic events -which occur whenever all their preconditions hold- and actions, which additionally require that an agent performs them. The agent's choice to perform an action is given by the `performs(S,A,T)` predicate, which represents the agent's 'free will', and is autonomous in that it itself depends on no preconditions. The new model also indexes each time-dependent predicate to a simulation, which associates theses predicates to a scenario. From an original simulation s_0, indexing serves to investigate hypothetical alternatives for modelling scenario-based properties of causality.

The domains used here are $\mathcal{S}$, $\mathcal{D}$, $\mathcal{T}$, $\mathcal{F}$, $\mathcal{U}$, $\mathcal{X}$, $\mathcal{A}$, $\mathcal{O}$, $\mathcal{I}$ and $\mathcal{E}$ corresponding respectively to simulations, agents, time points, positive fluents, automatic events, action names, actions, omissions, volitions and events.

We denote domains using cursive capitals and corresponding variables using print capitals, and use these domains to denote sets of predicates or functions. For instance, if `p` is a predicate or function of arity 1, $\mathtt{p}(\mathcal{F})$ denotes the set $\{\mathtt{p}(F), F \in \mathcal{F}\}$.

Planning Context. A *planning context* is composed of domain dependent *event specifications* and *initial situation* facts; it is denoted by Ctx. *Event specifications* define existing events, as well as their preconditions and effects, respectively given by `prec(F,E)` and `effect(E,F)`. Priorities between events ensure the precedence of one event over another when both are possible, and are given by `priority(E1,E2)`. The *initial situation* is composed of fluents that are true initially, denoted by `initially(F)`.

Event Motor. An *event motor* is a set of domain independent axioms governing the dynamics of a scenario. We here present a concise event motor adapted from [6]. The evolution of fluents is traced by the `holds(S,F,T)` predicate, which states that F is true at T in S. A fluent holds at T is S if it was initiated by an event occurrence at T-1 in S; a fluent true at T in S continues to hold until the occurrence of an event that terminates it, unless it is non-inertial, in which case it holds at T only. If a positive fluent does not hold at T in S, then its negation does. Hereunder is a translation in ASP. The complete source code with proof of concept is downloadable [1].

```
holds(S,F,1):-initially(F),sim(S).
holds(S,F,T+1):-occurs(S,E,T),effect(E,F),posFluent(F).
holds(S,F,T+1):-holds(S,F,T),not nonInertial(F),
    {occurs(S,E,T):effect(E,neg(F))}0,time(T).
holds(S,neg(F),T):-not holds(S,F,T),sim(S),posFluent(F),time(T).
```

The evolution of events is traced by the `occurs(S,E,T)` predicate, which states that E occurs at T in S. An event is complete when all its preconditions hold; an automatic event is possible when it is complete; an action is possible when it is both complete and performed; any event occurs when it is possible and not overtaken.

```
complete(S,E,T):-
    {not holds(S,F,T):prec(F,E)}0,sim(S),event(E),time(T).
possible(S,E,T):-complete(S,E,T),event(E),not action(E).
possible(S,A,T):-complete(S,A,T),performs(S,A,T),action(A).
overtaken(S,E1,T):-
    possible(S,E1,T),possible(S,E2,T),priority(E2,E1),E1!=E2.
occurs(S,E,T):-possible(S,E,T),not overtaken(S,E,T).
```

The union of these sets of axioms constitutes the *event motor*. The union of this *event motor* with the *planning context Ctx* is called the *basic action model* and is denoted by $\mathbb{A}^{b}_{Ctx}$.

2.2 Modelling Omissions

The meaningful fact of omitting to act only occurs when *acting is possible*. One cannot omit to act if there is no act to omit [39]. As such, we state that an omission occurs when an action is possible, not overtaken, and not performed. Given an action `act(D,X)`, its omission is denoted by `omit(D,X)`. We define the *overturn* operator ov by ov(`act(D,X)`) =`omit(D,X)` and ov(`omit(D,X)`) =`act(D,X)`. An omission O inherits the priority features of its corresponding action $ov(O)$. To prevent the same omission from occurring repeatedly, an action is not considered possible just after having been omitted, even while its preconditions are still true. In terms of dynamics, an omission has no effect on the current state of the world, such that all fluents, including non-inertial ones, that were true before its occurrence remain so after it. It does however have causal - as opposed to operational- effects that are opposite to those of the corresponding action. It makes true those fluents the action would terminate and terminates those the action would initiate (this supposes well defined actions that do not make true what is already true). For example, omitting to turn the light off initiates the fact that the light is still on and terminates the possibility that it be off. To take omissions into account, the *event motor* is updated by adding the following rules and removing the previous definition of `complete(S,E,T)`. Given a *planning context Ctx*, the *action model* obtained by replacing the *event motor* of section 2.1 by its update is called the *action model with omissions* and denoted by $\mathbb{A}^{o}_{Ctx}$.

```
complete(S,U,T):-{not holds(S,F,T):prec(F,U)}0,sim(S),auto(U),time(T).
complete(S,act(D,X),T):-{not holds(S,F,T):prec(F,act(D,X))}0,
```

[1] https://github.com/FBerreby/Aamas2018

```
    not occurs(S,omit(D,X),T-1),sim(S),action(act(D,X)),time(T).
complete(S,omit(D,X),T):-complete(S,act(D,X),T),
    not performs(S,act(D,X),T),not overtaken(S,act(D,X),T).
priority(omit(D,X),E):-priority(act(D,X),E).
holds(S,F,T+1):-occurs(S,omit(D,X),T),holds(S,F,T),nonInertial(F).
effect(omit(D,X),F):-effect(act(D,X),neg(F)).
effect(omit(D,X),neg(F)):-effect(act(D,X),F),posFluent(F).
```

2.3 Scenarios and Trace

To formally define meaningful sets of performed actions, we introduce the notion of scenarios.

Definition 2.1. Given a *planning context* Ctx, a *scenario* of Ctx is defined as a couple (s, P) with $s \in \mathcal{S}$ and $P \subseteq$`performs(`s`,`$\mathcal{A}$`,`$\mathcal{T}$`)` a *set of performed actions*, such that (i) $\mathbb{A}^o_{Ctx} \cup P$ is consistent and (ii) $\forall$`performs(`s`,A,T)` $\in P$, $\mathbb{A}^o_{Ctx} \cup P \models$`occurs(`$s$`,A,T)`, where $\models$ denotes the classical skeptical entailment in ASP.

The set of all scenarios of a given *planning context* Ctx is denoted by Σ_{Ctx}. Given that we will not consider multiple planning contexts, we drop the Ctx subscript in the remainder, except for formal definitions. Given a scenario as input, the *action model with omissions* derives the evolution of fluents and the occurrences of events. We denote by $Th_{\mathcal{L}}(\Pi)$ the projection of sceptical consequences of a program Π on the set of predicates $\mathcal{L}$, i.e. $Th_{\mathcal{L}}(\Pi) = \{p, \Pi \models p\} \cap \mathcal{L}$.

Definition 2.2. Given a *planning context* Ctx, the *execution trace* of a scenario $\sigma = (s, P)$ is defined as $tr_{Ctx}(\sigma) = Th_{\mathcal{L}_{h,occ}}(\mathbb{A}^o_{Ctx} \cup P)$ where $\mathcal{L}_{h,occ}$ =`holds(`s`,`$\mathcal{F}$`,`$\mathcal{T}$`)`$\cup$`occurs(`s`,`$\mathcal{E}$`,`$\mathcal{T}$`)`.

Depending on the task, we may be interested in investigating the causal properties of a single scenario or of multiple ones concurrently. These scenarios each correspond to an original simulation s_0 that is to be compared against relevant alternatives as described bellow. The set of all possible scenarios in a given context can be generated by adding the following lines to $\mathbb{A}^o_{Ctx}$, with each resulting answer set constituting one scenario.

```
time(1..n).
0{performs(s,A,T):action(A)}1:-time(T).
:-performs(s,A,T),not complete(s,A,T).
```

To focus on interesting scenarios, we can then add constraints. For instance, a rule of the form '`:-toAvoid(E),occurs(s,E,T).`' can preclude simulations by targeting and banning events that they contain. These rules determine the features that a simulation must have in order to be considered at all. As such, they operate upstream of any reasoning on causality.

3 EVENT-BASED CAUSALITY

Philosophers traditionally distinguish two notions of causality, typically called *type causality* ("Speeding causes accidents") and *actual causality* ("The fact that Caitlyn sped caused her to have an accident today") [14]. This paper focuses on the second notion. A further distinction is also sometimes made within actual causality between what should be considered the true causes of an outcome from what should be considered background conditions [34][16]. Consider the question: "Was it Caitlyn, the car's horsepower, or the existence of a road that caused the accident?" To answer it, we may pick one of those options and argue for why it is salient, or we may consider these to all be causes, because they all participate in some way in the outcome. This paper does the latter, and here distinguishes different facets of such actual causality.

Table 1: Matrix of Causal Relations

	Strong	**Weak**
Supporting	Causes	Enables
Opposing	Prevents	Excludes

Moral responsibility is typically associated with the *occurrence* of events, such as the dropping of a bomb or the allocation of funds to disaster areas. Yet responsibility is equally a question of avoided harms; much good is done and much damage averted by such things as medical investment, early drug prevention or the regulation of wartime conduct. We therefore distinguish two conditions of *causal direction*, determined by the nature of the outcome as a produced event or an avoided one. *Supporting causality* regards events that make true the preconditions to other events, *opposing causality* regards events that terminate the preconditions to other events.

Making true the preconditions to an automatic event is different from making true the preconditions to a volition whose occurrence also depends on the independent choice of the agent to perform it or not. Though we may be fully responsible for our actions and the automatic events that we cause, we cannot be fully causally responsible for the choices of others -though, legally, we might. We therefore distinguish two conditions of *causal strength*, determined by the nature of the event that is at the end of the causal chain. *Strong causality* designates the kind of relationship where an event makes true, or terminates, the preconditions to an automatic event. *Weak causality* designates the kind of relationship where an event makes true, or terminates, the preconditions to a volition.

As such, **causes** denotes strong supporting causality, **prevents** denotes strong opposing causality, **enables** denotes weak supporting causality, and **excludes** denotes weak opposing causality. It follows that automatic events can only be caused or prevented, and volitions only enabled or excluded. For example, if John harms Sam, then John **causes** this harm; but if John tells Pat where Sam is and Pat harms Sam, then John **enables** Pat's action, while Pat **causes** the harm. The nature of the event at the beginning of the causal chain has no impact on the direction or strength of the relation. Actions, omissions and automatic events can uniformly assume each kind of causal relation with a particular outcome.

3.1 Supporting Causality

Definition 3.1. An event E *causes* a fluent F if E initiates F, and both obtain. A fluent F *causes* an automatic event U if F is a precondition to U, and both obtain. An event E *causes* an automatic event U if E causes a fluent F which causes U. We denote it by E $\longmapsto$ U.

Definition 3.2. A fluent F *enables* a volition I if F is a precondition to I, and both obtain. An event E *enables* a volition I if E causes a fluent F which enables I. We denote it by E $\longmapsto$ I. Since automatic events and volitions are exclusive, this notation does not overlap with the previous one.

Causing is 'transparently' transitive, meaning that an event that causes an automatic event that itself causes, prevents, enables or excludes a third event assumes the relation that exists between the first two events. We refrain from imbuing the other types of causal relations with transitive powers, as assigning such powers demands further philosophical positioning. For example, it is not obvious what concept characterises the relation between an event that enables an action and another event that is prevented by this same action - does the first event simply prevent the second, or should a specific concept of ‘enabling to prevent’ be applied?

Modelling Supporting Causality. Defining causality on the basis of the Event Calculus architecture affords us with a functional trace of causal paths and allows us to dynamically assess causal relationships. We define causing in the following way. `r(S,causes,E,T,U)` indicates that E, which happened at T in S, causes the automatic event U. The referenced time point denotes the time at which occurred the first event within a causal chain.

```
posRel(causes;enables).
r(S,causes,E,T,F):-occurs(S,E,T),effect(E,F),holds(S,F,T+1).
r(S,causes,F,T,U):-holds(S,F,T),prec(F,U),occurs(S,U,T),auto(U).
r(S,R,E1,T1,E2):-r(S,causes,E1,T1,C),
    r(S,R,C,T2,E2),event(E1;E2),T2>T1,posRel(R).
```

We then use the definition and transitive power of causing to model enabled actions and omissions; the two definitions only depart in the last section between the last fluent preceding the caused or enabled event, and the event itself. The fact that an agent does or does not perform the action made complete by the truth-values of fluents determines whether it is an action or the corresponding omission that has been enabled; the enabling of an omission is derived from the completion of an action that is not performed. `r(S,enables,E,T,I)` indicates that E, which happened at T in S, enables the volition I.

```
r(S,enables,F,T,A):-holds(S,F,T),prec(F,A),occurs(S,A,T),action(A).
r(S,enables,F,T,omit(D,X)):-
    holds(S,F,T),prec(F,act(D,X)),occurs(S,omit(D,X),T).
```

The moral significance of enabling other people’s actions is easy to envision: we gain praise if we give money to a charity that dispenses medical supplies; we are to blame if we knowingly give a cocktail to an alcoholic person. Yet it can also be significant to make true the conditions for actions that are not performed. In such cases, the moral appraisal of the enabling agent and the omitting agent will typically contradict. If the charity in question fails to dispense medical supplies even though it received support, its culpability increases, but our goodwill remains. If the alcoholic person refrains from drinking, even though we made it easy for them to do so, they might gain additional praise, but our culpability remains.

3.2 Opposing Causality

Definition 3.3. An event E *prevents* an automatic event U if: (a) E terminates a fluent that is a precondition to U or to another automatic event which would cause U; (b) all other preconditions to U hold; (c) U does not occur. We denote it by E $\longmapsto \overline{U}$.

Definition 3.4. An event E *excludes* a volition I if: (a) E terminates a fluent that is a precondition to I or to an automatic event which would enable I; (b) all other preconditions to I hold; (c) I does not become complete. We denote it by E $\longmapsto \overline{I}$.

Modelling Opposing Causality. Opposing causality makes different computational demands than supporting causality. To model it, we define a number of prior predicates. `hyp(F1,F2)` denotes that F1 is a *hypothetical cause* of F2 if a causal link (made up of automatic events and fluents) exists between these two fluents. This predicate says nothing about the actual state of the world, i.e., about whether this causal link has been instantiated. It corresponds to *type* causality. `transTerm(S,E,F,T)` denotes that E *transterminates* F if E terminates F or terminates another fluent that is a hypothetical cause of F. This allows for indirect cases where E affects a non-contiguous fluent. It corresponds to the (a) clause of the definition of *prevents* and *excludes*. `canArise(S,E)` denotes that E occurred at some point in time in S. We also consider that if an action or an omission occurs, its corresponding omission or action also can arise in that simulation. Relative to volitions, this predicate serves to identify the fact that the *choice* of acting or omitting to perform an action has occurred. Relative to automatic events, it simply identifies their occurrence. `relevant(S,E1,T,E2)`, via `irrelevant(S,E1,T1,E2,T2)`, excludes the cases in which E1 transterminates a precondition to E2, but where at least one other precondition to E2 is missing that has not itself been transterminated by E1 in S. It preserves us from considering that something has been avoided when it wasn’t actually about to happen. For example, we do not want to say that a collision was prevented by our stopping of a car if there wasn’t anyone on the road to collide with.

```
hyp(F1,F2):-prec(F1,U),effect(U,F2),auto(U).
hyp(F1,F3):-hyp(F1,F2),hyp(F2,F3).
canArise(S,E):-occurs(S,E,T).
canArise(S,omit(D,X)):-occurs(S,act(D,X),T).
canArise(S,act(D,X)):-occurs(S,omit(D,X),T).
transTerm(S,E,F,T):-occurs(S,E,T),effect(E,neg(F)).
transTerm(S,E,F2,T):-transTerm(S,E,F1,T),hyp(F1,F2),posFluent(F2).
irrelevant(S,E1,T1,E2,T2):-
    transTerm(S,E1,F1,T1),not holds(S,F2,T2),prec(F1,E2),prec(F2,E2),
    not transTerm(S,E1,F2,T1),T2>T1,time(T2).
relevant(S,E1,T1,E2):-transTerm(S,E1,F1,T1),prec(F1,E2),
    not irrelevant(S,E1,T1,E2,T2),T2>T1,time(T2).
```

We can now define the pivot predicates for opposing causality. `r(S,prevents,E,T,U)` states that E *prevents* U at T in S, and `r(S,excludes,E,T,A)` states that E *excludes* A at T in S. An event that excludes an action also excludes its corresponding omission.

```
negRel(prevents;excludes).
r(S,prevents,E,T,U):-transTerm(S,E,F,T),prec(F,U),relevant(S,E,T,U),
    not canArise(S,U),auto(U).
r(S,excludes,E,T,A):-transTerm(S,E,F,T),prec(F,A),relevant(S,E,T,A),
    not canArise(S,A),action(A).
r(S,excludes,E,T,omit(D,X)):-r(S,excludes,E,T,act(D,X)).
```

3.3 Causal Trace

Let $\mathbb{C}^e$ be the set of all axioms presented in this section, called the *event-based causal model*. We define a causal trace as follows.

Definition 3.5. Given a *planning context Ctx*, the *causal trace* of a scenario $\sigma = (s, P)$ is defined as $ctr_{Ctx}(\sigma) = Th_{\mathcal{L}_r}(\mathbb{A}^o_{Ctx} \cup \mathbb{C}^e \cup P)$ where $\mathcal{L}_r$ =r($s, \mathcal{R}, \mathcal{E}, \mathcal{T}, \mathcal{E}$) with $\mathcal{R}$ = {`causes`, `enables`, `prevents`, `excludes`}.

We say that a scenario σ *verifies* a causal relation between two events if this relation belongs to the causal trace of σ. We denote it by $\sigma \models (\phi \longmapsto \psi)$ where ψ can respectively be ε or $\overline{\varepsilon}$ depending on whether the causal relation is a supporting or an opposing one (with ε an event). Therefore, whenever $\psi = \overline{\varepsilon}$, $\overline{\psi}$ denotes ε. Throughout, we call ϕ the *affector* and ψ the *end-state*.

4 SCENARIO-BASED CAUSALITY

Causality as it is modelled above is blind. This means that it is not concerned with the context in which the causal relationship occurs; in particular it is not concerned with the other causal relationships that hold in the situation. However, when ascribing responsibility to an agent, context can be determining. For example, homicides are characterised in severely different ways by virtue of the context in which they happened: as murder, manslaughter, or assisted suicide. Though it does not account for all aspects of it, one powerful way to investigate context is to submit the relationship between two events to counter-factual and conditional tests. The assisted suicide of a terminally ill patient is typically less reprehensible than other forms of homicide because, had the act of killing not occurred, it is assumed that the patient would have died anyway. The *existence of a terminal illness*, though external to the causal chain between the two events *act of killing* and *death of the patient*, influences how we view and morally appraise this causal chain, because it influences its counter-factual assessment. Empirically, counter-factual thinking has widely been shown to play an important role in moral reasoning (eg. [36][10][26]). Because ethical responsibility pertains to agents' choices, we consider only actions and omissions, rather than automatic events, as potential affectors. We now turn to exploring three counter-factual properties in which the if-clause "ϕ is not true" is contrary to fact, and one conditional property in which the if-clause "ϕ is true" conforms to fact.

Given a volition $\phi \in \mathcal{I}$ and a scenario $\sigma \in \Sigma(\phi, t)$, we denote by $\Sigma^{\sigma \to t}$ the completion of σ from t, which is defined as the set of all scenarios containing exactly the same set of actions performed strictly before t. If ϕ is an action (resp. omission), $\Sigma^{\sigma \to t, \phi}$ is the set of all scenarios of $\Sigma^{\sigma \to t}$ that contain $\texttt{performs(s,}\phi\texttt{,}t\texttt{)}$ (resp. do not contain $\texttt{performs(s,}ov(\phi)\texttt{,}t\texttt{)}$) and reciprocally $\Sigma^{\sigma \to t, \overline{\phi}}$ is the set of all scenarios of $\Sigma^{\sigma \to t}$ that do not contain $\texttt{performs(s,}\phi\texttt{,}t\texttt{)}$ (resp. contain $\texttt{performs(s,}ov(\phi)\texttt{,}t\texttt{)}$) .

Given an event $\varepsilon \in \mathcal{E}$, we denote the set of scenarios in which ε occurs at time t as $\Sigma(\varepsilon, t)$, and the set of scenarios in which ε does not occur at time t as $\Sigma(\overline{\varepsilon}, t)$. We can then define the set of scenarios in which ε occurs at least once (resp. not at all) after time t as $\Sigma_{t+}(\varepsilon) = \bigcup_{t_2 \in \mathcal{T}, t_2 \geq t} \Sigma(\varepsilon, t_2)$ (resp. $\Sigma_{t+}(\overline{\varepsilon}) = \bigcap_{t_2 \in \mathcal{T}, t_2 \geq t} \Sigma(\overline{\varepsilon}, t_2)$).

Simple counter-factual validity. Is ψ counter-factually dependent on ϕ? In other words, if ϕ had not been true, but all else had remained equal, would ψ also not have been true?

Definition 4.1. Given $\sigma = (s_0, P) \in \Sigma(\phi, t)$ such that $\sigma \models \phi \longmapsto \psi$, $\phi \longmapsto \psi$ is counter-factually valid iff $ov(\sigma_0, \phi, t) \in \Sigma_{t+}(\overline{\psi})$, where $ov(\sigma_0, \phi, t) \in \Sigma^{\sigma \to t, \overline{\phi}}$ is a unique completion of σ that overturns ϕ while keeping all other performed actions of σ that remain possible (see 4.1 to build it).

Cruciality. Was ϕ the only way to bring about ψ? In other words, if ϕ had not been true, was there any other possible volition that could make ψ true?

Definition 4.2. Given $\sigma = (s_0, P) \in \Sigma(\phi, t)$ such that $\sigma \models \phi \longmapsto \psi$, ϕ is *crucial* to ψ iff $\Sigma^{\sigma \to t, \overline{\phi}} \bigcap \Sigma_{t+}(\psi) = \varnothing$

Extrinsic Necessity. Was ψ necessary? In other words, if ϕ had not been true, would ϕ have necessarily been true anyway?

Definition 4.3. Given $\sigma = (s_0, P) \in \Sigma(\phi, t)$ such that $\sigma \models \phi \longmapsto \psi$, ψ is *extrinsically necessary* relative to ϕ iff $\Sigma^{\sigma \to t, \overline{\phi}} \bigcap \Sigma_{t+}(\overline{\psi}) = \varnothing$

Elicited Necessity. Does ϕ make ψ necessary? In other words, knowing ϕ is true, was there any possible later volition that could have stopped ϕ from being true?

Definition 4.4. Given $\sigma = (s_0, P) \in \Sigma(\phi, t)$ such that $\sigma \models \phi \longmapsto \psi$, *$\phi$ makes ψ necessary* if: $\Sigma^{\sigma \to t, \phi} \bigcap \Sigma_{t+}(\overline{\psi}) = \varnothing$

The remaining case set $\Sigma^{\sigma \to t, \phi} \bigcap \Sigma_{t+}(\psi)$ is trivially never empty because σ belongs to it. It should be noted that these properties can have diverging impacts on responsibility attribution: simple counter-factual validity, cruciality and elicited necessity have the tendency to heighten the responsibility of an agent's volition upon the outcome, where extrinsic necessity tends to diminish it.

4.1 A Tree of Simulations

To model the above properties, it is necessary to generate the hypothetical simulations from which we will infer them. This happens in two steps. First, to test for simple counter-factual validity, we generate simulations in which we overturn each affector ϕ in s_0 and nothing else, s_0 being the simulation corresponding to the scenario $\sigma_0 = (s_0, P)$ whose causal relation we investigate. Second, to test for the three other properties, we generate simulations that model every possible combination of volitions in the domain.

Modelling $ov(\sigma_0, \phi, t)$. For every ϕ that occurs in s_0 at time t, a child simulation is produced. It is referred to as $\texttt{ov(}s_0\texttt{,}\phi\texttt{,}t\texttt{)}$ and corresponds to the $ov(\sigma_0, \phi, t)$ scenario defined in definition 4.1. We say that it *emanates* from s_0. Since $ov(\sigma_0, \phi, t) \in \Sigma^{\sigma \to t, \overline{\phi}}$, we need to ensure ϕ is overturned in this new simulation. If it is an action, it will not occur because by default it will not be performed. If it is an omission, the agent performs the corresponding action. All actions other than ϕ that were performed at any time in the parent are performed in the child, unless they have been made incomplete by the overturning of ϕ. If new actions become possible, they will not be performed, for agent volition would be an external addition. Omissions are considered the default.

```
emanates(0,T,T):-occurs(0,I,T),volition(I).
sim(S2):-emanates(S1,S2,T).
performs(S2,A,T1):-
    emanates(S1,S2,T2),performs(S1,A,T1),complete(S2,A,T1),T1!=T2.
performs(S2,act(D,X),T):-occurs(S1,omit(D,X),T),emanates(S1,S2,T).
```

These steps ensure that we produce the first level of a tree of simulations emanating from what is now the root node, s_0.

Modelling All Possible Scenarios. Testing for the other three properties means that we must model all possible scenarios, i.e. each possible combination of volitions in the domain, and then look inside them. The first level of the tree of simulations generated above already contains the set of scenarios in which the volitions in s_0 have been overturned. The next step consists in creating new simulations for every volition that becomes possible in these first level scenarios. These second level scenarios may themselves contain possible volitions that will initiate further simulations until all options have been exhausted. This expands the tree, creating as many levels as there are volitions in the causal chain that has the most volitions.

Each child simulation is named using a number in which every figure references the time of occurrence of each overturned volition that composes it. For example, $s_2 4$ is a second level simulation in which two volitions were overturned, the first at time 2 in s_0 and the second at time 4 in s_2. Because only one volition is possible per time, this process successfully indexes all the simulations descending from s_0 without redundancy or crossover.

```
simIndex(1..99).
emanates(S2,S2*10+T2,T2):-emanates(S1,S2,T1),occurs(S2,I,T2),
    T1<T2,simIndex(S2),volition(I).
```

An example. Paul sees a car wreck where the driver is harmed. He aims to do some good, so only simulations in which some good was done will be considered as s_0, as discussed in 2.3. Take s_0 to contain 3 actions: performing first (aid) at T1, taking the driver to (safety) at T2, (calling) for help at T3 -enabling the driver to reach full recovery. (safety) is only possible once (aid) is performed -if not the driver dies so there is no taking them to safety. (call) is only possible once both (aid) and (safety) are performed. If (aid) and (safety) are performed but not (call), the driver only partially recovers for lack of prompt treatment, but this makes possible a new action, muscle (rehab). If he performs it, *then* he can reach full recovery. The tree of simulations will then be: s_1 omit(aid); s_2 aid, omit(safety); s_3 aid, safety, omit(call), auto(partRec), omit(rehab); s_{35} aid, safety, omit(call), auto(partRec), rehab, auto(fullRec).

4.2 Simple Counter-factual Validity

case 1 [c.f. validity] If the state had granted him asylum status, he would not have committed suicide.
case 2 [¬ c.f. validity] If the state had given him asylum status, he would have committed suicide anyway.

When we examine ours and other people's decisions, it is common to wonder what would have happened had the opposite decision been made. This is particularly true when the decision leads to an important outcome: What if I hadn't gone to the party and failed to meet the love of my life? What if he had abstained from driving in his drunken state? Moreover, ensuring that the outcome is counter-factually dependent on the decision yields a certain level of authority upon the causal relation. Indeed, if the outcome had been the same in the absence of the decision, then it might weaken the claim that the latter is responsible for the former.

Logicians have elaborated numerous counter-factual analyses of causation, most famously by appealing to the idea of a 'nearest possible world' [35] or sets of such worlds [22]. The claim is that a counter-factual is valid only if the world contradicting the affector that is the most similar to the actual world also contradicts the end-state. However, these proposals have met problems in giving a coherent account of the term 'nearest' [25]. Here, we take a simplifying stance and consider the nearest possible world to be the scenario in which the affector to an end-state is overturned (the overturned affector being the antecedent of the counter-factual).

This simple counter-factual test is one in which we imagine the affector to be negated, but where *all else is equal*, meaning that the scenario in which ϕ is negated is identical to the original one except for the negation of ϕ and all that for which ϕ is necessary. As such, if counter-factual validity affirms that $\neg\phi$ leads to $\neg\psi$, it does not affirm that $\neg\phi$ necessarily leads to $\neg\psi$. Though ψ *doesn't* occur in the absence of ϕ, it is not the case that it *couldn't* occur: if agents had behaved differently, it might have. The person in case 1 might still have committed suicide while having asylum status if, say, a personal tragedy had befallen them; reversely, the person in case 2 might have been saved by medication.

Modelling Simple Counter-factual Validity. We have $\phi \longmapsto \psi$ in s_0, and we seek to establish whether ψ is true in $ov(\sigma_0, \phi, t)$. Ensuring that an event-based causal relation already holds between ϕ and ψ means that unlike purely counter-factual accounts of causality, we only submit to this test pairs of events that are already shown to be somewhat causally related. This lightens the computational load. The `posRel(R)` and `negRel(R)` predicates distinguish supporting and opposing causal relations. Relative to supporting causality, we state that E is not counter-factually dependent on I if: (a) I causes or enables E in s_0; (b) E can arise in the simulation where I is overturned. Relative to opposing causality, we state that E is not counter-factually dependent on I if: (a) I prevents or excludes E in s_0; (b) E never arises in the simulation where I is overturned. From these rules, we derive the relations that are counter-factually valid.

```
notValid(0,R,I,T,E):-r(0,R,I,T,E),emanates(0,S,T),
    canArise(S,E),volition(I),event(E),posRel(R).
notValid(0,R,I,T,E):-r(0,R,I,T,E),emanates(0,S,T),
    not canArise(S,E),volition(I),event(E),negRel(R).
valid(0,R,I,T,E):-r(0,R,I,T,E),not notValid(0,R,I,T,E),
    volition(I),event(E).
```

4.3 Cruciality

case 3 [cruciality] Only his doctor could have noticed he had started using drugs and stopped it early on.
case 4 [¬ cruciality] Any one of his relatives could have noticed he had started using drugs and stopped it early on.

When ascribing responsibility, it can be important to know how many people, or events, have the power to lead to a particular outcome. For instance, if many people each had the capacity to bring about some desirable end but each failed to do so, blame might be shared. If only one person had this capacity, blame can only ever be attributed to them. This distinction may then influence how we characterise or weigh this blame. If there were many potential determinants, then we may consider that the outcome was easier to reach than if there was just one, meaning that the group failed more strongly than a unique determinant would have. Reversely, we may think that being crucial to an outcome yields an exclusive form of power which confers a particular moral status on the agent or their action, as is for example the case with the presidential power of pardon. The knowledge of such power may even affect an agent's own actions, for instance by pushing them to act more carefully.

Modelling Cruciality. We have $\phi \longmapsto \psi$ in s_0 and we seek to establish whether there exists a simulation in which ψ is true but ϕ isn't. It should be noted that an end-state which results from a unique crucial affector is determined by a *single causal chain*, rather than by a *single affector*. Indeed, If E1 causes E4 by passing through events E2 and E3, and no other events exist that can cause E4, then E1, E2 and E3 each uniquely determine E4, because they are part of a unique causal chain. As such, ϕ here represents a causal chain rather than a single element, and testing for cruciality means searching for a new causal chain leading to ψ.

We define the prior predicate `laterSim(exc,I,S)`, which selects for each I occurring at T in s_0 all the simulations corresponding to scenarios in $\Sigma^{\sigma_0 \to T,\bar{I}}$, i.e. simulations in which I does not occur and that are, up to T, identical to s_0 ('exc' standing for '*exclusive* of I').

```
differentHistory(T2,S1,S2):-occurs(S1,I,T1),not occurs(S2,I,T1),
    sim(S2),volition(I),time(T2),T1<T2.
differentHistory(T,S2,S1):-differentHistory(T,S1,S2).
laterSim(exc,act(D,X),S):-occurs(0,act(D,X),T),
    occurs(S,omit(D,X),T),not differentHistory(T,0,S).
laterSim(exc,omit(D,X),S):-occurs(0,omit(D,X),T),
    occurs(S,act(D,X),T),not differentHistory(T,0,S).
```

Relative to supporting causality, we then state that I is not crucial to E if: (a) I causes or enables E in s_0; (b) there is a later simulation exclusive of I in which E can arise. Relative to opposing causality, we state that I is not crucial to E if: (a) I prevents or excludes E in s_0; (b) there is a later simulation exclusive of I in which E cannot arise. From these rules, we infer relations of cruciality.

```
notCrucial(0,R,I,T,E):-r(0,R,I,T,E),laterSim(exc,I,S),
    canArise(S,E),volition(I),event(E),posRel(R).
notCrucial(0,R,I,T,E):-r(0,R,I,T,E),laterSim(exc,I,S),
    not canArise(S,E),volition(I),event(E),negRel(R).
crucial(0,R,I,T,E):-r(0,R,I,T,E),not notCrucial(0,R,I,T,E),
    volition(I),event(E).
```

4.4 Extrinsic Necessity and Elicited Necessity

We here discuss and model two properties of necessity.

case 5 [extrinsic necessity] If I hadn't picked the money up from the floor, someone else would have, and it would never have been returned to the owner.
case 6 [¬ extrinsic necessity] If I hadn't picked the money up from the floor, no one else would have, and it might have been returned to the owner.

It is a point of debate whether someone can be said to truly cause or be responsible for an outcome that would necessarily have happened had they not caused it themselves. Imagine that someone jaywalks across a highway, and a car hits them. If they hadn't hit them, another car necessarily would have. We typically would not hold the driver of the first car fully liable. Yet there are also cases where necessity seems insufficient to fully challenge responsibility. A mob member who commits a murder orchestrated by their boss might attempt to rationalise their act by claiming that if they hadn't done it, another mob member would have. Yet it would still seem right to condemn them for murder. We may additionally want the indictment of the mob boss. Whatever the situation, it is often important to investigate such circumstantial aspects.

case 7 [elicited necessity] Knowing no one else could come into the kitchen to turn it off, a cook lights the oven and then leaves. A fire starts.
case 8 [¬ elicited necessity] Thinking that the next shift baker would use the oven and then turn it off, a cook lights the oven then leaves. The baker fails to show up, and a fire starts.

Initiating a causal chain that could eventually lead to a particular outcome is very different from initiating a causal chain which ensures, regardless of what every agent might do, that the outcome occurs. Making an outcome necessary means there will be no 'going back', and no one else to hold accountable for intervening or failing to intervene before the outcome occurs. Even if it is not an outright intention, the knowledge of initiating a causal chain that cannot be challenged makes such a decision more taxing and significant. The known unavoidability of the outcome in case 7 heavily points in the direction of ill intention and liability. Reversely, as in case 8, we routinely initiate causal chains leading to a dangerous outcome because we know, or think, they will be broken before it occurs.

The properties of extrinsic and elicited necessity are closely linked: extrinsic necessity is the state of an event as necessary in the absence of one's volition, elicited necessity is the creation of that state by a volition. These two properties can be combined to generate four cases with sharply distinct ramifications:

case 9 [¬extrinsic, ¬elicited], **case 10 [extrinsic, elicited]**,
case 11 [¬extrinsic, elicited], **case 12 [extrinsic, ¬ elicited]**.

case 9 is a neutral case relative to these properties in that the end-state is avoidable both before and after the considered affector volition occurs. **case 10** is a case of absolute necessity in that the end-state is necessary whether the affector volition occurs or not (and not just if it does not, as in extrinsic necessity). Here, affector responsibility over the end-state is diminished. **case 11** is a case of true elicited necessity in which the previously avoidable end-state is imposed on the situation by the affector volition. Here, affector responsibility over the end-state is strengthened. Finally, **case 12** is somewhat counter-intuitive in that the end-state is only necessary in the absence of the affector volition, but not in its presence. As such, even if the affector volition indeed produces the end-state in s_0, it might rightly be seen as an attempt to avoid it rather than to produce it, in that it allows for the possibility of avoiding it where its absence would not. Affector responsibility is here open to debate.

Modelling Necessity. Pertaining to *extrinsic necessity*, we have $\phi \longmapsto \psi$ in s_0 and we seek to establish whether there exists a simulation in which neither ϕ nor ψ are true. This property characterises the end-state at the time when the affector volition becomes complete, by determining whether it will necessarily arise from this moment onward, regardless of what every agent does. To model it, we employ the `laterSim(exc,I,S)` predicate defined above. Pertaining to *elicited necessity*, we have $\phi \longmapsto \psi$ in s_0 and we seek to establish whether there exists a simulation in which ϕ is true and ψ isn't. We define the prior predicate `laterSim(inc,I,S)`, which selects for each I occurring at T in s_0 all the simulations corresponding to scenarios in $\Sigma^{\sigma_0 \to T,I}$, i.e. simulations in which I occurs and that are, up to T, identical to s_0.

Relative to supporting causality, we state that E is not extrinsically necessary relative to I (resp. I does not make E necessary) if: (a) I causes or enables E in s_0; (b) there exists a later simulation exclusive (resp. inclusive) of I in which E cannot arise. Relative to opposing causality, we state that E is not extrinsically necessary relative to I (resp. I does not make E necessary) if: (a) I prevents or excludes E in s_0; (b) there exists a later simulation exclusive (resp. inclusive) of I in which E can arise. From these rules, we infer relations of extrinsic necessity (resp. elicited necessity). As such, the two properties can be modelled together, each distinctly determined by the 'exc' and 'inc' conditions denoted by K.

```
laterSim(inc,I,S):-occurs(0,I,T),occurs(S,I,T),
    not differentHistory(T,0,S),volition(I).
notNecessary(0,R,K,I,T,E):-r(0,R,I,T,E),laterSim(K,I,S),
    not canArise(S,E),volition(I),event(E),posRel(R).
notNecessary(0,R,K,I,T,E):-r(0,R,I,T,E),laterSim(K,I,S),
    canArise(S,E),volition(I),event(E),negRel(R).
necessary(0,R,K,I,T,E):-r(0,R,I,T,E),not notNecessary(0,R,K,I,T,E),
```

Table 2: Dependencies

Crucial				¬ Crucial			
C.F. Valid		¬ C.F. Valid		C. F. Valid		¬ C.F. Valid	
¬ E.N.	E.N.	¬ E.N.	E.N.	¬ E.N.	E.N.	¬ E.N.	E.N.
a				b		c	d

```
    volition(I),event(E),condition(K).
```

4.5 Dependencies

The counter-factual properties defined above are partially interdependent, and in part mutually exclusive. For example, if ϕ causes ψ while the later was extrinsically necessary, then necessarily the relation is counter-factually invalid. This results from the fact that the properties are more or less stringent: the more stringent they are, the more they might command upon the others. These dependencies are summarised in table 2 with properties organised in descending order of stringency, from cruciality to simple counter-factual validity to extrinsic necessity (E.N.). The instances a, b, c and d represent the four possible combinations of counter-factual properties, impossible ones being greyed out.

5 DISCUSSION AND RELATED WORKS

The presented work is pertinent to two distinct but related domains of research, computational ethics and the logical analysis of causality. Pertaining to the first, there exist a number of engaging attempts to model ethical reasoning (e.g. [2][3][7][12][20][30]). However, they all have in common that they do not represent causality explicitly, such that an action and its consequences are not dynamically linked; causal relations are stated rather than inferred. This eclipses the underlying dynamics that make up causal reasoning, cutting short the possibility to provide a justifiable account of ethical responsibility on its basis. The absence of expressively powerful causal rules also limits the applicability and scope of these models, meaning that they typically require an entirely new program to model each new scenario, even when there are common features. To our knowledge, this work is also the first in the domain to assimilate actions, omissions and automatic events within plans of actions for modelling responsibility. This greatly increases the capacity to model true-to-life scenarios, but also permits the analysis of ethically critical distinctions that have often been overlooked.

Going back to Hume [18], the notion of causality has been widely analysed and defined, appraised variously through counter-factual, probabilistic or structural models (e.g. [9][21][22][29][31]). Works in this domain almost all treat causality as an all-or-nothing concept[2], yet, when attributing responsibility, it is essential to be able to reason in terms of degrees as well as remain sensitive to context. This is reflected in legal concepts of responsibility and has been demonstrated empirically [40][32]. This paper does not aim to give definitions of causality in accordance with this, rather it provides a framework for modelling the properties that might make up such definitions. We here give two examples of how this might be done relative to existing accounts given by logicians and philosophers.

We appeal to the properties of necessity to model the definition of *active causal responsibility* using STIT logic given in [24], informally defined as 'agent i is actively causally responsible for bringing about ψ if i sees to it that ψ and ψ is not inevitable in the sense that it is true regardless of what every agent does'.

```
loriniACR(0,R,I,T,U):-necessary(0,R,inc,I,T,U),
    notNecessary(0,R,exc,I,T,U),auto(U).
```

Relative to causal strength, the distinction between causing and enabling can be used to model diverging models of liability. While Moore, as well as Hart and Honore [16][28], claim that one shouldn't be said to cause an evil when one's act merely enables the action of another agent to cause that evil, Gardner upholds that complicity in the wrongdoing of another constitutes a cause of this wrongdoing [13]. His account can be simply modelled by generating specific transitive powers for enabling.

```
r(S,causes,E,T1,U):-r(S,enables,E,T1,I),r(S,causes,I,T2,U),
    event(E),auto(U),T1<T2.
```

More generally, all the distinctions made in the paper can be used to fine tune consequentialist theories of ethics, by helping to determine what indeed are the relevant causes and consequences of agent's volitions.

6 CONCLUSION

This work adapts and builds on the Event Calculus to allow the modelling of causal properties that are fundamental to ethical reasoning and to the attribution of ethical responsibility. It investigates and represents ethically significant distinctions relative to the ways in which agents may act upon the world: through actions or omissions that might lead to produced or avoided outcomes. It then defines causal properties derived from the exploration of alternative versions of an original scenario that help to decipher, strengthen or diminish the attribution of responsibility for a particular outcome. This permits the generation of rules with valuable expressive power which equip agents with the capacity to decide upon their decisions, but also to reason over other agent's actions.

We envision a number of future avenues. We first aim to explore ways of expressing intentionality, as it is so far only handled implicitly. The properties of scenario-based causality are also to be supplemented by auxiliary properties with further explanatory power. In particular, we aim to look into *preemption*, when one cause is replaced by another had the first failed to obtain, and *over-determination*, when there are more causes present than are necessary to produce an outcome. Identifying such cases may impact on responsibility attribution, and it also matters to identify *which agent* is at the origin of the preemption or over-determination. For example, if an agent makes necessary an end-state that is also over-determined by the volition of another agent, then two agents may share responsibility. If it is the same agent that causes the over-determination, then this relief is cancelled. In continuity with this target, we aim to extend tracing responsibility over single volitions to tracing it over sets of volitions, in particular sets emanating from a single agent and from coalitions of agents. This will pave the way towards integrating the model within a multi-agent system, so as to more fully exploit its potential to enable cooperation or collective intelligence. Finally, we defer to future research an extension of the model that will handle notions of uncertainty in causation.

[2] A noteworthy exception being Chockler and Halpern's probabilistic structural model which investigates other aspects of degrees of causality, such as the responsibility shared between multiple agents that participate in a single outcome [8][15].

REFERENCES

[1] M Anderson and S Anderson. 2011. *Machine ethics*. Cambridge University Press.

[2] Michael Anderson, Susan Leigh Anderson, and Chris Armen. 2006. MedEthEx: a prototype medical ethics advisor.

[3] Ronald Arkin. 2009. *Governing lethal behavior in autonomous robots*. CRC Press.

[4] H Beebee, C Hitchcock, and P Menzies. 2009. *The Oxford handbook of causation*. Oxford University Press.

[5] Jonathan Bennett and Jonathan Francis Bennett. 1998. *The act itself*. Oxford University Press.

[6] Fiona Berreby, Gauvain Bourgne, and Jean-Gabriel Ganascia. 2017. A Declarative Modular Framework for Representing and Applying Ethical Principles. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 96–104.

[7] Selmer Bringsjord and Joshua Taylor. 2012. The divine-command approach to robot ethics. *Robot Ethics: The Ethical and Social Implications of RoboticsâĂŹ, MIT Press, Cambridge, MA* (2012), 85–108.

[8] Hana Chockler and Joseph Y Halpern. 2004. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research* 22 (2004), 93–115.

[9] John David Collins, Edward Jonathan Hall, and Laurie Ann Paul. 2004. *Causation and counterfactuals*. MIT Press.

[10] Kai Epstude and Neal J Roese. 2008. The functional theory of counterfactual thinking. *Personality and Social Psychology Review* 12, 2 (2008), 168–192.

[11] Philippa Foot. 1985. Morality, action, and outcome. *Morality and objectivity* (1985), 23–38.

[12] Jean-Gabriel Ganascia. 2015. Non-monotonic resolution of conflicts for ethical reasoning. In *A Construction Manual for Robots' Ethical Systems*. Springer, 101–118.

[13] John Gardner. 2007. Moore on Complicity and Causality. *U. Pa. L. Rev. PENNumbra* 156 (2007), 432.

[14] Joseph Y Halpern. 2015. A Modification of the Halpern-Pearl Definition of Causality.. In *IJCAI*. 3022–3033.

[15] Joseph Y Halpern. 2016. *Actual causality*. MIT Press.

[16] Herbert Lionel Adolphus Hart and Tony Honoré. 1985. *Causation in the Law*. OUP Oxford.

[17] J Horty. 1997. Nonmonotonic foundations for deontic logic. In *Defeasible deontic logic*. Springer.

[18] D Hume. 2012. *A treatise of human nature*. Courier Corporation.

[19] Shelly Kagan. 1989. The limits of morality. (1989).

[20] R Kowalski. 2011. *Computational logic and human thinking: how to be artificially intelligent*. Cambridge University Press.

[21] David Lewis. 1974. Causation. *The journal of philosophy* 70, 17 (1974), 556–567.

[22] David Lewis. 2013. *Counterfactuals*. John Wiley & Sons.

[23] V Lifschitz. 2008. What Is Answer Set Programming?. In *AAAI*, Vol. 8. 1594–1597.

[24] Emiliano Lorini, Dominique Longin, and Eunate Mayor. 2013. A logical analysis of responsibility attribution: emotions, individuals and collectives. *Journal of Logic and Computation* 24, 6 (2013), 1313–1339.

[25] Peter Menzies. 2001. Counterfactual theories of causation. (2001).

[26] Simone Migliore, Giuseppe Curcio, Francesco Mancini, and Stefano F Cappa. 2014. Counterfactual thinking in moral judgment: an experimental study. *Frontiers in psychology* 5 (2014).

[27] R Miller and M Shanahan. 2002. Some alternative formulations of the event calculus. In *Computational logic: logic programming and beyond*. Springer, 452–490.

[28] Michael S Moore. 2007. Causing, aiding, and the superfluity of accomplice liability. *University of Pennsylvania Law Review* 156, 2 (2007), 395–452.

[29] Judea Pearl. 2003. Causality: models, reasoning and inference. *Econometric Theory* 19, 675-685 (2003), 46.

[30] L M Pereira and A Saptawijaya. 2007. Modelling morality with prospective logic. In *Progress in Artificial Intelligence*. Springer, 99–111.

[31] Luís Moniz Pereira and Ari Saptawijaya. 2017. Counterfactuals, logic programming and agent morality. In *Applications of Formal Philosophy*. Springer, 25–53.

[32] John V Petrocelli, Elise J Percy, Steven J Sherman, and Zakary L Tormala. 2011. Counterfactual potency. *Journal of personality and social psychology* 100, 1 (2011), 30.

[33] Murray Shanahan. 1999. The event calculus explained. In *Artificial intelligence today*. Springer, 409–430.

[34] Steven Sloman, Aron K Barbey, and Jared M Hotaling. 2009. A causal model theory of the meaning of cause, enable, and prevent. *Cognitive Science* 33, 1 (2009), 21–50.

[35] Robert C Stalnaker. 1968. A theory of conditionals. In *Ifs*. Springer, 41–55.

[36] Philip E Tetlock, Penny S Visser, Ramadhar Singh, Mark Polifroni, Amanda Scott, Sara Beth Elson, Philip Mazzocco, and Phillip Rescober. 2007. People as intuitive prosecutors: The impact of social-control goals on attributions of responsibility. *Journal of Experimental Social Psychology* 43, 2 (2007), 195–209.

[37] Achille C Varzi. 2007. Omissions and causal explanations. (2007).

[38] Bruce Vermazen and Merrill B Hintikka. 1985. Essays on Davidson. (1985).

[39] Bernard Weiner. 1995. *Judgments of responsibility: A foundation for a theory of social conduct*. guilford Press.

[40] Ro'i Zultan, Tobias Gerstenberg, and David A Lagnado. 2012. Finding fault: causality and counterfactuals in group attributions. *Cognition* 125, 3 (2012), 429–440.