On Automating the Doctrine of Double Effect

Naveen Sundar Govindarajulu and **Selmer Bringsjord**

Rensselaer Polytechnic Institute, Troy, NY

{naveensundarg,selmer.bringsjord}@gmail.com

Abstract

The **doctrine of double effect** ($\mathcal{DDE}$) is a long-studied ethical principle that governs when actions that have both positive and negative effects are to be allowed. The goal in this paper is to automate $\mathcal{DDE}$. We briefly present $\mathcal{DDE}$, and use a first-order modal logic, the **deontic cognitive event calculus**, as our framework to formalize the doctrine. We present formalizations of increasingly stronger versions of the principle, including what is known as the **doctrine of *triple* effect**. We then use our framework to successfully simulate scenarios that have been used to test for the presence of the principle in human subjects. Our framework can be used in two different modes: One can use it to build $\mathcal{DDE}$-compliant autonomous systems from scratch; or one can use it to verify that a given AI system is $\mathcal{DDE}$-compliant, by applying a $\mathcal{DDE}$ layer on an existing system or model. For the latter mode, the underlying AI system can be built using any architecture (planners, deep neural networks, bayesian networks, knowledge-representation systems, or a hybrid); as long as the system exposes a few parameters in its model, such verification is possible. The role of the $\mathcal{DDE}$ layer here is akin to a (dynamic or static) software verifier that examines existing software modules. Finally, we end by sketching initial work on how one can apply our $\mathcal{DDE}$ layer to the STRIPS-style planning model, and to a modified POMDP model. This is preliminary work to illustrate the feasibility of the second mode, and we hope that our initial sketches can be useful for other researchers in incorporating $\mathcal{DDE}$ in their own frameworks.

1 Introduction

The **doctrine of double effect** ($\mathcal{DDE}$) is a long-studied ethical principle that enables adjudication of ethically "thorny" situations in which actions that have both positive and negative effects appear unavoidable for autonomous agents [McIntyre, 2014]. Such situations are commonly called *moral dilemmas*. The simple version of $\mathcal{DDE}$ states that such actions, performed to "escape" such dilemmas, are allowed

— provided that 1) the harmful effects are not intended; 2) the harmful effects are not used to achieve the beneficial effects (harm is merely a *side*-effect); and 3) benefits outweigh the harm by a significant amount. What distinguishes $\mathcal{DDE}$ from, say, naïve forms of consequentialism in ethics (e.g. act utilitarianism, which holds that an action is obligatory for an autonomous agent if and only if it produces the most utility among all competing actions) is that purely mental intentions in and of themselves, independent of consequences, are considered crucial (as condition 2 immediately above conveys). Of course, every major ethical theory, not just consequentialism, has its passionate proponents; cogent surveys of such theories make this plain (e.g. see [Feldman, 1978]). Even in machine ethics, some AI researchers have explored not just consequentialism and the second of the two dominant ethical theories, deontological ethics (marked by an emphasis on fixed and inviolable principles said by their defenders to hold no matter what the consequences of abrogating them), but more exotic ones, for example contractualism (e.g. see [Pereira and Saptawijaya, 2016b]) and even divine-command ethics (e.g. see [Bringsjord and Taylor, 2012]). $\mathcal{DDE}$ in a sense rises above philosophical debates about which ethical theory is preferred. The first reason is that empirical studies have found that $\mathcal{DDE}$ plays a prominent role in an ordinary person's ethical decisions and judgments [Cushman *et al.*, 2006]. For example, in [Hauser *et al.*, 2007], a large number of participants were asked to decide between action and inaction on a series of moral dilemmas, and their choices adhered to $\mathcal{DDE}$, irrespective of their ethical persuasions and backgrounds, and no matter what the order in which the dilemmas were presented. In addition, in legal systems, criminality requires the presence of malicious intentions [Fletcher, 1998], and $\mathcal{DDE}$ plays a central role in many legal systems [Allsopp, 2011; Huxtable, 2004].[1] Assuming that autonomous systems will be expected to adjudicate moral dilemmas in human-like ways, and to justify such adjudication, it seems desirable to seek science and engineering that allows $\mathcal{DDE}$, indeed even nuanced, robust versions thereof, to be quickly computed.

---

[1]On the surface, *criminal negligence* might seem to require no intentions. While that might be true, even in criminal negligence it seems rational to ask whether the negligence was accidental or something the "suspect" had control over. This suggests a milder form of intention, or something similar, but not exactly intention.

2 Prior Work

We quickly review prior rigorous modeling of $\mathcal{DDE}$. Mikhail in [Mikhail, 2011] presents one of the first careful treatments of the doctrine. While the presentation of the doctrine makes use of some symbolism, the level of formalization is not amenable to automation. [Bentzen, 2016] presents a model-theoretic formalization of a simple version of the doctrine. While this is an important first step, the calculus presented by Bentzen does not have any computational realization. However, there are two independent strands of research with implementations for $\mathcal{DDE}$: that of Berreby et al. [2015] and Pereira and Saptawijaya [2016a]; both use logic programming. Notably, while the Berreby et al. explicitly eschew counterfactuals for modeling $\mathcal{DDE}$, Pereira and Saptawijaya model $\mathcal{DDE}$ using counterfactuals. To our knowledge, both the projects present one of the first formal models of $\mathcal{DDE}$ that can be implemented.

It should be noted, however, that both of these formal systems are **extensional**, and it is well-known that when dealing with **intensional** states such as knowledge, belief, intention etc., extensional systems can quickly generate inconsistencies [Bringsjord and Govindarajulu, 2012] (see the appendix for more details). The expressivity challenge is both quantificational and intensional; this challenge is acute for the logic-programming paradigm, as opposed to one based — as is ours — on formal languages beyond first-order logic and its variants, and proof theories beyond resolution and its derivatives. In particular, $\mathcal{DDE}$ requires elaborate structures for quantification (including, inevitably, first-order numerical quantifiers such as $\exists^k : k \in \mathbb{R}$, since quantification over utilities is essential), and many intensional operators that range over quantifiers, starting with the epistemic ones. Needless to say, modeling and simulation at the propositional level, while truly excellent in the case of [Pereira and Saptawijaya, 2016a], is insufficiently expressive.

Among the many empirical experiments centered around $\mathcal{DDE}$, the one in [Malle *et al.*, 2015] deserves a mention. Malle *et al.* devise an experiment in which they place either a human or a robot as the central actor in a hypothetical $\mathcal{DDE}$ scenario, and study an external viewer's moral judgement of action or inaction by the human or robot. This study shows that humans view ethical situations differently when robots participate in such situations; and the study demonstrates the need for rigorous modeling of $\mathcal{DDE}$ to build well-behaved autonomous systems that function in $\mathcal{DDE}$-relevant scenarios.

3 The Calculus

In this section, we present the **deontic cognitive event calculus** ($\mathcal{DCEC}$). Dialects of this calculus have been used to formalize and automate highly intensional reasoning processes, such as the false-belief task [Arkoudas and Bringsjord, 2008] and *akrasia* (succumbing to temptation to violate moral principles) [Bringsjord *et al.*, 2014].[2] $\mathcal{DCEC}$ is a sorted (i.e. typed) quantified modal logic (also known as sorted first-order modal logic). The calculus has a well-defined syntax

---

[2]Arkoudas and Bringsjord [2008] introduced the general family of **cognitive event calculi** to which $\mathcal{DCEC}$ belongs.

and proof calculus; see [Bringsjord *et al.*, 2014]. The proof calculus is based on natural deduction [Gentzen, 1935], and includes all the introduction and elimination rules for first-order logic, as well as inference schemata for the modal operators and related structures. A snippet of $\mathcal{DCEC}$ is shown in the Appendix.

3.1 Syntax

First-order Fragment

The first-order core of $\mathcal{DCEC}$ is the *event calculus* [Mueller, 2006]. Though we use the event calculus, our approach is compatible with other calculi (e.g. the *situation calculus*) for modeling events and their effects.

Modal Fragment

The modal operators present in the calculus include the standard operators for knowledge **K**, belief **B**, desire **D**, intention **I**, etc. The general format of an intensional operator is $\mathbf{K}(a,t,\phi)$, which says that agent a knows at time t the proposition ϕ. Here ϕ can in turn be any arbitrary formula.

The calculus also includes a dyadic deontic operator **O**. The unary ought in standard deontic logic is known to lead to contradictions. Our dyadic version of the operator blocks the standard list of such contradictions, and beyond.[3]

3.2 Semantics

First-order Fragment

The semantics for the first-order fragment is the standard first-order semantics. The truth-functional connectives $\wedge, \vee, \rightarrow, \neg$ and quantifiers $\forall, \exists$ for pure first-order formulae all have the standard first-order semantics.

Modal Fragment

The semantics of the modal operators differs from what is available in the so-called Belief-Desire-Intention (BDI) logics [Rao and Georgeff, 1991] in many important ways. For example, $\mathcal{DCEC}$ explicitly rejects possible-worlds semantics and model-based reasoning, instead opting for a *proof-theoretic* semantics and the associated type of reasoning commonly referred to as *natural deduction* [Gentzen, 1935; Francez and Dyckhoff, 2010]. Briefly, in this approach, meanings of modal operators are defined via arbitrary computations over proofs, as we will soon see.

Reasoner (Theorem Prover)

Reasoning is performed through a novel first-order modal logic theorem prover, ShadowProver, which uses a technique called **shadowing** to achieve speed without sacrificing consistency in the system. Extant first-order modal logic theorem provers that can work with arbitrary inference schemata are built upon first-order theorem provers. They achieve the reduction to first-order logic via two methods. In the first method, modal operators are simply represented by first-order predicates. This approach is the fastest but can quickly lead to well-known inconsistencies as demonstrated in [Bringsjord and Govindarajulu, 2012]. In the second method, the entire proof theory is implemented intricately in first-order logic,

---

[3]A nice version of the list is given lucidly in [McNamara, 2010].

and the reasoning is carried out within first-order logic. Here, the first-order theorem prover simply functions as a declarative programming system. This approach, while accurate, can be excruciatingly slow. We use a different approach, in which we alternate between calling a first-order theorem prover and applying modal inference schemata. When we call the first-order prover, all modal atoms are converted into propositional atoms (i.e., shadowing), to prevent substitution into modal contexts. This approach achieves speed without sacrificing consistency. The prover also lets us add arbitrary inference schemata to the calculus by using a special-purpose language. While we use the prover in our simulations, describing the prover in more detail is out of scope for the present paper.[4]

4 Informal $\mathcal{DDE}$

We now informally but rigorously present $\mathcal{DDE}$. We assume we have at hand an ethical hierarchy of actions as in the deontological case (e.g. forbidden, neutral, obligatory); see [Bringsjord, 2017]. We also assume that we have a utility or goodness function for states of the world or effects as in the consequentialist case. For an autonomous agent a, an action α in a situation σ at time t is said to be $\mathcal{DDE}$-compliant *iff*:

C$_1$ the action is not forbidden (where we assume an ethical hierarchy such as the one given by Bringsjord [2017], and require that the action be neutral or above neutral in such a hierarchy);

C$_2$ The net utility or goodness of the action is greater than some positive amount γ;

C$_{3a}$ the agent performing the action intends only the good effects;

C$_{3b}$ the agent does not intend any of the bad effects;

C$_4$ the bad effects are not used as a means to obtain the good effects; and

C$_5$ if there are bad effects, the agent would rather the situation be different and the agent not have to perform the action. That is, the action is unavoidable.

See Clause 6 of Principle III in [Khatchadourian, 1988] for a justification of of **C$_5$**. This clause has not been discussed in any prior rigorous treatments of $\mathcal{DDE}$, but we feel **C$_5$** captures an important part of when $\mathcal{DDE}$ is normally used, e.g. in unavoidable ethically thorny situations one would rather not be present in. **C$_5$** is necessary, as the condition is subjunctive/counterfactual in nature and hence may not always follow from **C$_1$** − **C$_4$**, since there is no subjunctive content in those conditions. Note that while [Pereira and Saptawijaya, 2016a] model $\mathcal{DDE}$ using counterfactuals, they use counterfactuals to model **C$_4$** rather than **C$_5$**.

That said, the formalization of **C$_5$** is quite difficult, requiring the use of computationally hard counterfactual and subjunctive reasoning. We leave this aside here, reserved for future work.

---

[4]The prover is available in both Java and Common Lisp and can be obtained at: https://github.com/naveensundarg/prover. The underlying first-order prover is SNARK available at: http://www.ai.sri.com/~stickel/snark.html.

5 Formal $\mathcal{DDE}$

The formalization is straightforward given the machinery of $\mathcal{DCEC}$. Let Γ be a set of **background** axioms, which could include whatever the given autonomous agent under consideration knows about the world; e.g., its understanding of physics, knowledge and beliefs about other agents and itself, etc. The particular **situation** that might be in play, e.g., *"the autonomous agent is driving,"* is represented by a formula σ. We use ground fluents for effects.

We assume that we have a utility function μ that maps from fluents and times to real-number utility values. μ needs to be defined only for ground fluents:

$$\mu : \mathsf{Fluent} \times \mathsf{Moment} \to \mathbb{R}$$

Good effects are fluents with positive utility, and bad effects are fluents that have negative utility. Zero-utility fluents could be neutral fluents (which do not have a use at the moment).

5.1 Defining *means* $\triangleright$

The standard event calculus and $\mathcal{DCEC}$ don't have any mechanism to say when an effect is used as a **means** for another effect. While we could employ a first-order predicate and define axiomatically when an effect is used as a means for another effect, we take a modal approach that does not require any additional axioms beyond what is needed for modeling a given situation. Intuitively, we could say an effect e_1 is a mere side effect for achieving another effect e_2 if by removing the entities involved in e_1 we can still achieve e_2; otherwise we say e_1 is a means for e_2. Our approach is inspired by Pollock's [1976] treatment, and while similarities can be found with the approach in [Pereira and Saptawijaya, 2016a], we note that our definition requires at least first-order logic. Given a fluent f, we denote by $\odot$ the set of all constants and function expressions in f. For example:

$$\odot \left(hungry(jack) \right) = \{jack\}$$
$$\odot \left(married(jack, sister(mary)) \right) = \{jack, sister(mary), mary\}$$

We need one more definition: the state of the world without a given set of entities. Let $\otimes(\Gamma, \theta)$, where Γ is a set of formulae and θ is a set of ground terms, be defined as below:

$$\otimes \left(\Gamma, \theta \right) = \left\{ \psi \in \Gamma \mid \psi \text{ does not contain any term in } \theta \right\}$$

Note that the above definition relies on the **Unique Names Assumption** commonly used in most formulations of the event calculus. This assumption ensures that every object in the domain has at most one name or expression referring to it. If this assumption does not hold, we can have the following slightly more complicated definition for $\otimes$.

$$\otimes \left(\Gamma, \theta \right) = \left\{ \psi \in \Gamma \; \middle| \; \begin{array}{l} \psi \text{ does not contain any } s \text{ such that} \\ \exists t \in \theta : \Gamma \vdash s = t \end{array} \right\}$$

We introduce a new modal operator $\triangleright$, *means*, that says when an effect is a means for another effect.

$$\triangleright : \text{Formula} \times \text{Formula} \rightarrow \text{Formula}$$

The meaning of the operator is defined computationally below. The definition states that, given Γ, a fluent f holding true at t_1 causes or is used as a means for another fluent g at time t_2, with $t_2 > t_1$, *iff* the truth condition for g changes when we remove formulae that contain entities involved in f. While this definition is far from perfect, it suffices as a first cut and lets us simulate experimental scenarios that have been used to test $\mathcal{DDE}$'s presence in humans. (Three other similar definitions hold when we look at combinations of fluents holding and not holding.) The equation below follows (Note that $\vdash$ is non-monotonic, as it includes the event calculus):

$$\Gamma \vdash \triangleright\Big(Holds(f,t_1), Holds(g,t_2)\Big)$$

$$iff$$

$$\left\{ \begin{array}{c} \Gamma \vdash t_2 > t_1 \wedge \\ \left[\begin{array}{c} \Gamma \vdash Holds(f,t_1) \wedge \\ \Gamma \vdash Holds(g,t_2) \end{array} \right] \Rightarrow \Big[\Gamma - \otimes(\Gamma, \odot(f)) \vdash \neg Holds(g,t_2) \Big] \end{array} \right\}$$

For example, let e_1 be "*throwing a stone **s** at a window **w**"* and e_2 be "*the window **w** getting broken.*" We can see that e_2 is not just a mere side effect of e_1, and the definition works, since, if the stone is removed, e_2 wouldn't happen. This definition is not perfect. For instance, consider when there are common objects in both the events: the intuitiveness breaks down (but the definition still works). We might for example let e_1 be "*hitting a window **w** with a bat **b**.*" If the window and bat are not present, e_2 would not happen.

5.2 The Formalization

Note that the $\mathcal{DDE}(\Gamma, \sigma, a, \alpha, t, H)$ predicate defined below, though defined using $\mathcal{DCEC}$, lies outside of the formal language of $\mathcal{DCEC}$. While $\mathcal{DDE}$ is not fully formalized in $\mathcal{DCEC}$, the individual clauses $\mathbf{F}_1 - \mathbf{F}_4$ are. This is how we can verify the conditions in the simulations described later. It is trivial to define a new symbol and formalize the predicate in $\mathcal{DCEC}$: $\mathcal{DDE}1 \Leftrightarrow \mathbf{F}_1 \wedge \mathbf{F}_2 \wedge \mathbf{F}_3 \wedge \mathbf{F}_4$.

What is not trivial, we concede, is how this works with other modalities. For example, can we efficiently derive $\mathbf{K}(a,t_1,\mathbf{K}(b,t_2,\mathcal{DDE}(\Gamma,\sigma,a,\alpha,t,H)))$ given some other formulae Γ? This could be difficult because the predicate's definition below involves provability, and one has to be careful when including a provability predicate.

That said, for future work, we plan on incorporating this within an extended dialect of $\mathcal{DCEC}$. One immediate drawback is that while we can have a *system*-level view of whether an action is $\mathcal{DDE}$-sanctioned, agents themselves might not know that. For example, we would like to able to write down *"a knows that b knows that c's action is $\mathcal{DDE}$-sanctioned."*

Given the machinery defined above, we now proceed to the formalization. Assume, for any action type α carried out by an agent a at time t, that it initiates the set of fluents $\alpha_I^{a,t}$, and terminates the set of fluents $\alpha_T^{a,t}$. Then, for any action α

taken by an autonomous agent a at time t with background information Γ in situation σ, the action adheres to the doctrine of double effect up to a given time horizon H, that is $\mathcal{DDE}(\Gamma, \sigma, a, \alpha, t, H)$ *iff* the conditions below hold:

Formal Conditions for $\mathcal{DDE}$

$\mathbf{F}_1$ α carried out at t is not forbidden. That is:

$$\Gamma \nvdash \neg\mathbf{O}\Big(a,t,\sigma,\neg happens(action(a,\alpha),t)\Big)$$

$\mathbf{F}_2$ The net utility is greater than a given positive real γ:

$$\Gamma \vdash \sum_{y=t+1}^{H}\left(\sum_{f \in \alpha_I^{a,t}} \mu(f,y) - \sum_{f \in \alpha_T^{a,t}} \mu(f,y) \right) > \gamma$$

$\mathbf{F}_{3a}$ The agent a intends at least one good effect. ($\mathbf{F}_2$ should still hold after removing all other good effects.) There is at least one fluent f_g in $\alpha_I^{a,t}$ with $\mu(f_g,y) > 0$, or f_b in $\alpha_T^{a,t}$ with $\mu(f_b,y) < 0$, and some y with $t < y \leq H$ such that the following holds:

$$\Gamma \vdash \left(\begin{array}{c} \exists f_g \in \alpha_I^{a,t} \ \mathbf{I}\Big(a,t,Holds(f_g,y)\Big) \\ \vee \\ \exists f_b \in \alpha_T^{a,t} \ \mathbf{I}\Big(a,t,\neg Holds(f_b,y)\Big) \end{array} \right)$$

$\mathbf{F}_{3b}$ The agent a does not intend any bad effect. For all fluents f_b in $\alpha_I^{a,t}$ with $\mu(f_b,y) < 0$, or f_g in $\alpha_T^{a,t}$ with $\mu(f_g,y) > 0$, and for all y such that $t < y \leq H$ the following holds:

$$\Gamma \nvdash \mathbf{I}\Big(a,t,Holds(f_b,y)\Big) \text{ and}$$

$$\Gamma \nvdash \mathbf{I}\Big(a,t,\neg Holds(f_g,y)\Big)$$

$\mathbf{F}_4$ The harmful effects don't cause the good effects. Four permutations, paralleling the definition of $\triangleright$ above, hold here. One such permutation is shown below. For any bad fluent f_b holding at t_1, and any good fluent f_g holding at some t_2, such that $t < t_1, t_2 \leq H$, the following holds:

$$\Gamma \vdash \neg\triangleright\Big(Holds(f_b,t_1), Holds(f_g,t_2)\Big)$$

$\mathbf{F}_5$ This clause requires subjunctive reasoning. The current formalization ignores this stronger clause. There has been some work in computational subjunctive reasoning that we hope to use in the future; see [Pollock, 1976].

Doctrine of Triple Effect

The doctrine of triple effect ($\mathcal{DTE}$) was proposed in [Kamm, 2007] to account for scenarios where actions that are viewed as permissible by most philosophers and deemed as such by empirical studies (e.g. the switch action in the third scenario in [Hauser *et al.*, 2007]) are not sanctioned by $\mathcal{DDE}$, as they involve harm being used as a means to achieve an action. $\mathcal{DTE}$ allows such actions as long as the harm is not explicitly intended by the agent. Note that our version of $\mathcal{DDE}$

subsumes $\mathcal{DTE}$ through condition $\mathbf{C}_4$.

6 Scenarios

The trolley problems are quite popular in both philosophical and empirical studies in ethics. Hauser *et al.* [2007] found empirical support that $\mathcal{DDE}$ is used by humans, courtesy of experiments based on a set of trolley problems. They use a set of 19 trolley problems in their experimentation, and describe in detail four of these. We consider the first two of these problems in our study here. The problem scenarios are briefly summarized below; common to both this setup: There are two tracks $track_1$ and $track_2$. There is a trolley loose on $track_1$ heading toward two people P_1 and P_2 on $track_1$; neither person can move in time. If the trolley hits them, they die. The goal is to save this pair.[5]

Scenario 1 There is a switch that can route the trolley to $track_2$. There is a person P_3 on $track_2$. Switching the trolley to $track_2$ will kill P_3. Is it okay to switch the trolley to $track_2$?

Scenario 2 There is no switch now, but we can push P_3 onto the track in front of the trolley. This action will damage the trolley and stop it; it will also kill P_3. Is it okay to push P_3 onto the track?

$\mathcal{DDE}$-based analysis tells us it is okay to switch the trolley in **Scenario 1**, as we are killing the person merely as a side effect of saving P_1 and P_2. In **Scenario 2**, similar analysis tells us it is not okay to push P_3, because we are using that person as a means toward our goal.

7 Simulations

At the core of our simulation is a formalization of the basic trolley scenario based on the event calculus. We use a discrete version of the event calculus, in which time is discrete, but other quantities and measures, such as the utility function, can be continuous. We have the following additional sorts: Trolley and Track. We also declare that the Agent and the Trolley sorts are subsorts of a Moveable sort, the instances of which are objects that can be placed on tracks and moved. We use the following additional core symbols:

$position$: Moveable $\times$ Track $\times$ Number $\rightarrow$ Fluent

$dead$: Agent $\rightarrow$ Fluent

$onrails$: Trolley $\times$ Track $\rightarrow$ Fluent

$switch$: Trolley $\times$ Track $\times$ Track $\rightarrow$ ActionType

$push$: Agent $\times$ Track $\times$ Number $\rightarrow$ ActionType

The utility function μ is defined as follows:

$$\mu(f,t) = \begin{cases} -1 & \text{if } f \equiv dead(P) \\ 0 & \text{otherwise} \end{cases}$$

We set the threshold γ at 0.5. The simulation starts at time $t = 0$ with the only trolley, denoted by $trolley$, on $track_1$. We have an event-calculus trajectory axiom shown below as part of Γ:

---

[5]For computational purposes, the exact number of persons is not important as long as it is greater than one.

$\forall t$: Trolley, $track$: Track, s : Moment

$$\left[Trajectory\Big(onrails(t,track),s,position(t,track,\Delta),\Delta\Big) \right]$$

The above axiom gives us the trolley's position at different points of time. Γ also includes axioms that account for non effects. For example, in the absence of any actions, we can derive:

$$\Gamma \vdash Holds\Big(position\big(trolley,track_1,23\big),23\Big)$$

We also have in the background Γ a formula stating that in the given trolley scenario the agent ought to save both P_1 and P_2. Ideally, while we would like the agent to arrive at this obligation from a more primitive set of premises, this setup is closer to experiments with human subjects in which they are asked explicitly to save the persons. Note the agent performing the action is simply denoted by I, and let the time of the test be denoted by *now*.

$$\mathbf{O}\left(I,now,\sigma_{trolley}, \begin{bmatrix} \neg \exists t : \text{Moment } Holds\,(dead(P_1,t)) \wedge \\ \neg \exists t : \text{Moment } Holds\,(dead(P_2,t)) \end{bmatrix} \right)$$

Given that the agent knows that it is now in situation $\sigma_{trolley}$, and the agent believes that it has the above obligation, we can derive from $\mathcal{DCEC}$'s inference schemata what the agent intends:

$$\left\{ \begin{array}{l} \mathbf{K}\Big(I,now,\sigma_{trolley}\Big), \\ \mathbf{B}\left(I,now,\mathbf{O}\left(\begin{array}{l} I,now,\sigma_{trolley}, \\ \begin{bmatrix} \neg \exists t : \text{Moment } Holds\Big(dead(P_1,t)\Big) \\ \wedge \\ \neg \exists t : \text{Moment } Holds\Big(dead(P_2,t)\Big) \end{bmatrix} \end{array} \right) \right), \\ \mathbf{O}\left(I,now,\sigma_{trolley}, \begin{bmatrix} \neg \exists t : \text{Moment } Holds\,(dead(P_1,t)) \wedge \\ \neg \exists t : \text{Moment } Holds\,(dead(P_2,t)) \end{bmatrix} \right) \end{array} \right\}$$

$$\vdash \mathbf{I}\left(I,now, \begin{bmatrix} \neg \exists t : \text{Moment } Holds\Big(dead(P_1,t)\Big) \wedge \\ \neg \exists t : \text{Moment } Holds\Big(dead(P_2,t)\Big) \end{bmatrix} \right)$$

In both the simulations, P_1 is at position 4 and P_2 is at position 5 on $track_1$. In **Scenario 1**, P_3 is at position 3 on $track_2$, and the train can be switched from position 3 on $track_1$ to position 0 on $track_2$.

In **Scenario 2**, we push P_3 onto position 3 on $track_1$. The total number of formulae and run times for simulating the two scenarios with and without the actions are shown below. Note these are merely event-calculus simulation times. These are then used in computing $\mathcal{DDE}(\Gamma,\sigma,a,\alpha,t,H)$. The event-calculus simulation helps us compute $\mathbf{F}_2$.

Scenario	$|\Gamma|$	Simulation Time (s)	
		No action	*Action performed*
Scenario 1	39	0.591	1.116
Scenario 2	38	0.602	0.801

ShadowProver was then used to verify that $\mathbf{F}_1$, $\mathbf{F}_{3a}$, and $\mathbf{F}_{3b}$ hold. Both the scenarios combined take 0.57 seconds for $\mathbf{F}_1$, $\mathbf{F}_{3a}$, and $\mathbf{F}_{3b}$. The scenarios differ only in $\mathbf{F}_4$. The pushing action fails to be $\mathcal{DDE}$-compliant due to $\mathbf{F}_4$. For verifying that $\mathbf{F}_4$ holds in **Scenario 1** and doesn't hold in **Scenario 2**, it takes 0.49 seconds and 0.055 seconds, respectively.[6]

8   On Operationalizing the Principle

Given the above formalization, it's quite straightforward to build logic-based systems that are $\mathcal{DDE}$-compliant.[7] But how do we apply the above formalization to existing models and systems that are not explicitly logic-based? We lay down a set of conditions such models must satisfy to be able to verify that they are $\mathcal{DDE}$-compliant. We then sketch how we could use $\mathcal{DDE}$ in two such modified systems: a STRIPS-like planner and a POMDP type model.

The problem now before us is: Given a system and a utility function, can we say that the system is $\mathcal{DDE}$-compliant? No, we need more information from the system. For example, we can have two systems in the same situation, the same utility functions and same set of available actions.[8] One system can be $\mathcal{DDE}$-compliant while the other is not. For example, assume that we have two autonomous driving systems d_1 and d_2. Assume that d_2 has learned to like killing dogs and intends to do so if possible during its normal course of operation. While driving, both come across a situation where the system has to hit either a human or a dog. In this scenario, d_1's action to hit the dog would be $\mathcal{DDE}$-compliant while d_2's action will not be. Therefore, the formalization requires that we have access to an agent's intentions at all times.

One common objection to requiring that intentions be separate from utilities states that utilities can be used to derive intentions. This is mistaken: it is not always possible to derive intentions from a utility function. For example, there might be a state that has high utility but the agent might not intend to realize that state, as it could be out of reach for that agent (low perceived probability of success).

For instance, winning a million dollars (w) has high utility, but most rational agents might not intend w, as they know this event is (alas) out of their reach. This holds for similar high-utility states.

At a minimum, we believe utility and perceived probability of success go into an agent's intentions. This seems to align with the human case when we are looking at motivations, i.e. *expectancy-value* theory. How motivations could transform into intentions is another open research question.

---

[6]All the axioms for the two simulations, ShadowProver, and the combined $\mathcal{DDE}$ implementation can be obtained here: https://goo.gl/9KU2L9.

[7]For examples of logic-based systems in pure first-order logic, see [Mueller, 2006].

[8]Where does a utility function come from? The obvious way to get a utility function seems to be to learn such a function. There are good arguments that such learning can be very hard [Arnold *et al.*, 2017]. For now, we are not concerned with how such a utility function is given to us. For exposition and economy assume that it already exists.

8.1   Requirements

Practically speaking, there is a spectrum of systems that our techniques will be dealing with. At one end, we will encounter systems that are complete black boxes taking in percepts from the environment and spitting out actions. Since $\mathcal{DDE}$ requires us to look at intentions of systems, such black-box systems will be impossible to verify. We can of course ask the system to output its intention through language as one of its possible actions, but this means that we are relying on the system's honest reporting of its internal states. At the other end of the spectrum, we have complete white-box systems. We can be fully confident that we can get what the system intends, believes, knows, etc. at any point in time. Verifying such systems is possible, in theory at least. While we don't know what kind of shape autonomous systems will take and where they will fall in the spectrum, we can explicitly list information we need from such systems before we can start the verification process. One such specification follows.

> **Gray Box Requirement**
>
> Given any autonomous system a, at any point of time t, we should *at least* be able to assert the following, if true, in order to verify that it is $\mathcal{DDE}$-compliant:
>
> 1. The system's intentions: $(\neg)\mathbf{I}(a,t,\phi)$
> 2. Prohibitions: $\neg\mathbf{O}(a,t,\sigma,\neg\phi)$

How would we go about applying the formalization to other formal systems? We very briefly sketch two examples.

STRIP-like Planner

We first look at a STRIPS-style planning system. Briefly, a STRIPS-style planner has a set of actions $\{a_i\}$ and a set of states $\{s_i\}$. The states are nothing but sets of formulae or atoms. The individual formulae would be our effects. Each action a has a set of preconditions $pre(a)$, a set of formulae that should hold in a given state to execute that action in that state. After executing an action a in a state s, the new state is given by $s \cup additions(a) - deletions(a)$. The planner is given an explicit goal ϕ. This means that we know $(\neg)\mathbf{I}(a,t,\phi)$ trivially. If we have an ethical hierarchy for the available set of actions, we then satisfy the gray-box requirement. What is then needed is a definition for $\rhd$, an effect used as means for another effect. The formalism gives us one possible way to define $\rhd$. A plan ρ is nothing but a sequence of actions. Given a plan ρ, we say an effect e_1 is used as means for another effect e_2, if $e_1 \in pre(a_1)$, a_1 is an action in the plan and $e_2 \in additions(a_2)$, and a_1 comes before a_2.

POMDP-derived System

Partially observable Markov decision process (POMDP) models have been quite successful in a large number of domains. It is highly likely that some of the first autonomous systems might be based on POMDPs. We note that in such models, the only goal is to maximize a reward function. Another issue is that states are atomic. In order to discern between good and bad effects, we would need states to be decomposed into smaller components. One possible approach could use *factored markov decision processes*, which are MDPs in which states are represented as a mapping m from a

set of state variables $\Theta = \{s_1, s_2, \ldots, s_n\}$ to a set of values $\mathcal{V}$. Here the utility and reward function could be defined on the assignments; i.e., $reward(s) = \sum \mu(s_i \rightsquigarrow \nu)$, where μ assigns a utility value to a particular assignment of a state variable. Additionally, the formalism could specify one or more goal states that the model seeks to attain while maximizing the reward along the way, giving us $(\neg)\mathbf{I}(a, t, \phi)$.

9 Heirarchies of Doctrines

Our formalization, summarized in the equation below, gives rise to multiple hierarchies of the doctrine. We discuss some of the hierarchies below.

$$\mathcal{DDE}(\Gamma, \sigma, a, \alpha, t, H) \Leftrightarrow \mathbf{F}_1 \wedge \mathbf{F}_2 \wedge \mathbf{F}_3 \wedge \mathbf{F}_4$$

Horizon One obvious knob in the above equation is the horizon H. Increasing H will give us stronger versions of the doctrine. Since our formalization is in first-order modal logic, the horizon need not be finite: we could set the horizon to infinity, $H = \omega$, and still obtain a tractable model, as long as we carefully develop our formalization.[9]

Agent Generality Instead of just checking whether an action at a given time is $\mathcal{DDE}$-compliant, we could ask whether an autonomous agent a in a given situation σ will be $\mathcal{DDE}$-compliant at all times. This gives us the following condition:

$$\forall \alpha : \mathsf{ActionType}, t : \mathsf{Moment}. \; \mathcal{DDE}(\Gamma, \sigma, a, \alpha, t, H)$$

Situation Generality In the hierarchy above, the quantification was over objects. We could ask whether an autonomous agent would be $\mathcal{DDE}$-compliant in all situations. That would correspond to a quantification over formulae (see centered formula immediately below), something not supported in the version of $\mathcal{DCEC}$ used herein.

$$\forall \sigma : \mathsf{Formula}, \alpha : \mathsf{ActionType}, t : \mathsf{Moment}. \mathcal{DDE}(\Gamma, \sigma, a, \alpha, t, H)$$

Counterfactual Reasoning The presence or absence of counterfactual reasoning in $\mathbf{F}_5$ would correspond to a very strong version of the doctrine, but one that would also be very hard to automate in the general case. We note that there are hierarchies of counterfactual reasoning (see [Pollock, 1976]) that could correspond to hierarchies of versions of $\mathcal{DDE}$.

10 Conclusion

We now quickly summarize the chief contributions of the foregoing, and end by presenting future lines of work. Our primary contribution is the presentation of a novel computational logic, or cognitive calculus, in which important versions of $\mathcal{DDE}$ are formalized. As a part of this calculus, we formalized an effect being used as a means for another effect via the modal operator $\triangleright$. We also supplied an informal but rigorous version, $\mathbf{C}_1 - \mathbf{C}_4$, of the doctrine itself, from which we built our formalization $\mathbf{F}_1 - \mathbf{F}_4$. Included in this formalization is the clause $\mathbf{C_5}/\mathbf{F_5}$, which requires subjunctive and counterfactual reasoning, an aspect that hitherto has simply not been considered in any systematic treatment of $\mathcal{DDE}$. Our formalization subsumes the doctrine of *triple*

effect, $\mathcal{DTE}$; we have achieved the first computational simulations of the doctrine. A byproduct of these simulations is an event-calculus formalization of a demanding class of trolley problems (widely used in empirical and philosophical studies of ethics). We noted that our formalization gives rise to hierarchies of doctrines with varying strengths. Our readers can choose a particular strength doctrine that fits their needs.

Future work includes simulating more intricate "ethically thorny" scenarios. Despite our progress, we note that our formalization is devoid of any mechanisms for handling uncertainty, and we are in the process of extending our work to include reasoning based on probabilistic versions of $\mathcal{DCEC}$.[10] We also note that we have not said much about how our formalization could interact with an autonomous learning agent. We observe that even the possibility that such an intricate principle as $\mathcal{DDE}/\mathcal{DTE}$ is learnable using existing learning frameworks remains open to question [Arnold *et al.*, 2017]. In the short term, a guaranteed-to-be-fruitful but less ambitious area of development will be the deployment of our mechanization of $\mathcal{DDE}$ in existing systems, and adapting existing formal models, as briefly discussed above, to exploit this mechanization. Finally, we note that since we are using first-order (multi) modal logic, we will eventually run into efficiency issues, as even vanilla first-order logic's decision problem, $\Gamma \vdash \gamma$, is Turing-undecidable. There are a number of techniques to mitigate this issue. One approach is to exploit a library of commonly used proof patterns codified in a denotational proof language; see [Arkoudas and Bringsjord, 2008]. We are cautiously optimistic, as many formal enterprises outside of AI (e.g. software verification [Khasidashvili *et al.*, 2009] and formal physics [Stannett and Németi, 2014]) routinely face such challenges and surmount them.

Acknowledgements
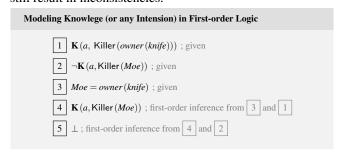
A Deontic Cognitive Event Calculus

We provide here a short primer on the deontic cognitive event calculus ($\mathcal{DCEC}$). A calculus is a set of axioms in a formal logic. For example, the event calculus is a set of axioms couched in first-order logic. $\mathcal{DCEC}$ is a set of axioms in sorted first-order modal logic (also known as sorted quantified modal logic) that subsumes the event calculus.

While first-order logic is an **extensional** system, modal logics are **intensional** systems. Note that there is a profound difference between intension vs. intention. One can have an intention to bring something about; this is traditionally captured by particular intensional operators. In other words, put concretely, the intention operator $\mathbf{I}$ is an intensional operator, but so is $\mathbf{D}$ for desire, $\mathbf{B}$ for believes, and $\mathbf{P}$ for perceives, etc.

---

[9]It's a well-known fundamental result that first-order logic can handle infinite models with a finite number of axioms; see e.g. Ch. 12 in [Boolos *et al.*, 2003].

[10]There exist probabilistic versions of the event calculus. We will leverage similar work.

$\mathcal{DCEC}$ is **intensional** in the sense that it includes intensional operators. Unfortunately, the situation is further confused by the fact that traditionally in philosophy of mind, intentionality means the so-called "aboutness" of some mental states, so that my belief that Melbourne is beautiful is in this sense intentional, while my mental state has nothing to do with intending something. Most logicians working in formal intensional systems believe that at least intensional logic is required to formalize intentional states [Zalta, 1988]. One simple reason is that using plain first-order logic leads to unsound inferences as shown below. In the inference below, we have an agent a that knows that the killer in a particular situation is the person that owns the knife. Agent a does not know that the *Moe* is the killer, but it's true that *Moe* is the owner of the knife. If the knowledge operator **K** is a simple first-order predicate, we will get the proof shown below, which produces a contradiction from sound premises. See [Bringsjord and Govindarajulu, 2012] for a sequence of stronger representation schemes in first-order logic for knowledge and belief that still result in inconsistencies.

Modeling Knowlege (or any Intension) in First-order Logic

1. $\mathbf{K}(a, \text{Killer}(owner(knife)))$; given

2. $\neg\mathbf{K}(a, \text{Killer}(Moe))$; given

3. $Moe = owner(knife)$; given

4. $\mathbf{K}(a, \text{Killer}(Moe))$; first-order inference from ③ and ①

5. $\bot$; first-order inference from ④ and ②

A.1 Syntax of Deontic Cognitive Event Calculus

$\mathcal{DCEC}$ is a sorted calculus. A sorted system can be thought of as being analogous to a typed single-inheritance programming language. We show below some of the important sorts used in $\mathcal{DCEC}$. Among these, the Agent, Action and ActionType sorts are not native to the event calculus.

Sort	Description
Agent	Human and non-human actors.
Time	The Time type stands for time in the domain. E.g. simple, such as t_i, or complex, such as $birthday(son(jack))$.
Event	Used for events in the domain.
ActionType	Action types are abstract actions. They are instantiated at particular times by actors. Example: eating.
Action	A subtype of Event for events that occur as actions by agents.
Fluent	Used for representing states of the world in the event calculus.

The figures below show the syntax and inference schemata of $\mathcal{DCEC}$. The syntax is quantified modal logic. Commonly used function and relation symbols of the event calculus are included. Particularly, note the following modal operators: **P** for perceiving a state, **K** for knowledge, **B** for belief, **C** for common knowledge, **S** for agent-to-agent communication and public announcements, **B** for belief, **D** for desire, **I** for intention, and finally and crucially, a dyadic deontic operator **O** that states when an action is obligatory or forbidden for agents. It should be noted that $\mathcal{DCEC}$ is one specimen in a *family* of easily extensible cognitive calculi. Since the semantics of $\mathcal{DCEC}$ is proof-theoretic, as long as a new construct has appropriate inference schemata, the extension is sanctioned.

Syntax

$S ::= \text{Object} \mid \text{Agent} \mid \text{ActionType} \mid \text{Action} \sqsubseteq \text{Event} \mid \text{Moment} \mid \text{Formula} \mid \text{Fluent}$

$$f ::= \begin{cases} action : \text{Agent} \times \text{ActionType} \rightarrow \text{Action} \\ initially : \text{Fluent} \rightarrow \text{Formula} \\ Holds : \text{Fluent} \times \text{Moment} \rightarrow \text{Formula} \\ happens : \text{Event} \times \text{Moment} \rightarrow \text{Formula} \\ clipped : \text{Moment} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Formula} \\ initiates : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Formula} \\ terminates : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Formula} \\ prior : \text{Moment} \times \text{Moment} \rightarrow \text{Formula} \end{cases}$$

$t ::= x : S \mid c : S \mid f(t_1, \ldots, t_n)$

$$\phi ::= \begin{cases} t : \text{Formula} \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \mathbf{P}(a,t,\phi) \mid \mathbf{K}(a,t,\phi) \mid \mathbf{C}(t,\phi) \\ \mathbf{S}(a,b,t,\phi) \mid \mathbf{S}(a,t,\phi) \mid \mathbf{B}(a,t,\phi) \mid \mathbf{D}(a,t,Holds(f,t')) \mid \mathbf{I}(a,t,\phi) \\ \mathbf{O}(a,t,\phi,(\neg)happens(action(a^*,\alpha),t')) \end{cases}$$

The figure below shows the inference schemata for $\mathcal{DCEC}$. $R_{\mathbf{K}}$ and $R_{\mathbf{B}}$ are inference schemata that let us model idealized agents that have their knowledge and belief closed under the $\mathcal{DCEC}$ proof theory. While normal humans are not dedcutively closed, this lets us model more closely how deliberate agents such as organizations and more strategic actors reason. (Some dialects of cognitive calculi restrict the number of iterations on intensional operators.) R_1 and R_2 state respectively that it is common knowledge that perception leads to knowledge, and that it is common knowledge that knowledge leads to belief. R_3 lets us expand out common knowledge as unbounded iterated knowledge. R_4 states that knowledge of a proposition implies that the proposition holds. R_5 to R_{10} provide for a more restricted form of reasoning for propositions that are common knowledge, unlike propositions that are known or believed. R_{12} states that if an agent s communicates a proposition ϕ to h, then h believes that s believes ϕ. R_{14} dictates how obligations get translated into intentions.

Inference Schemata

$$\frac{\mathbf{K}(a,t_1,\Gamma), \ \Gamma \vdash \phi, \ t_1 \leq t_2}{\mathbf{K}(a,t_2,\phi)} \ [R_{\mathbf{K}}] \qquad \frac{\mathbf{B}(a,t_1,\Gamma), \ \Gamma \vdash \phi, \ t_1 \leq t_2}{\mathbf{B}(a,t_2,\phi)} \ [R_{\mathbf{B}}]$$

$$\frac{}{\mathbf{C}(t,\mathbf{P}(a,t,\phi) \rightarrow \mathbf{K}(a,t,\phi))} \ [R_1] \qquad \frac{}{\mathbf{C}(t,\mathbf{K}(a,t,\phi) \rightarrow \mathbf{B}(a,t,\phi))} \ [R_2]$$

$$\frac{\mathbf{C}(t,\phi) \ t \leq t_1 \ldots t \leq t_n}{\mathbf{K}(a_1,t_1,\ldots \mathbf{K}(a_n,t_n,\phi)\ldots)} \ [R_3] \qquad \frac{\mathbf{K}(a,t,\phi)}{\phi} \ [R_4]$$

$$\frac{}{\mathbf{C}(t,\mathbf{K}(a,t_1,\phi_1 \rightarrow \phi_2)) \rightarrow \mathbf{K}(a,t_2,\phi_1) \rightarrow \mathbf{K}(a,t_3,\phi_2)} \ [R_5]$$

$$\frac{}{\mathbf{C}(t,\mathbf{B}(a,t_1,\phi_1 \rightarrow \phi_2)) \rightarrow \mathbf{B}(a,t_2,\phi_1) \rightarrow \mathbf{B}(a,t_3,\phi_2)} \ [R_6]$$

$$\frac{}{\mathbf{C}(t,\mathbf{C}(t_1,\phi_1 \rightarrow \phi_2)) \rightarrow \mathbf{C}(t_2,\phi_1) \rightarrow \mathbf{C}(t_3,\phi_2)} \ [R_7]$$

$$\frac{}{\mathbf{C}(t,\forall x. \ \phi \rightarrow \phi[x \mapsto t])} \ [R_8] \qquad \frac{}{\mathbf{C}(t,\phi_1 \leftrightarrow \phi_2 \rightarrow \neg\phi_2 \rightarrow \neg\phi_1)} \ [R_9]$$

$$\frac{}{\mathbf{C}(t,[\phi_1 \wedge \ldots \wedge \phi_n \rightarrow \phi] \rightarrow [\phi_1 \rightarrow \ldots \rightarrow \phi_n \rightarrow \psi])} \ [R_{10}]$$

$$\frac{\mathbf{S}(s,h,t,\phi)}{\mathbf{B}(h,t,\mathbf{B}(s,t,\phi))} \ [R_{12}] \qquad \frac{\mathbf{I}(a,t,happens(action(a^*,\alpha),t'))}{\mathbf{P}(a,t,happens(action(a^*,\alpha),t))} \ [R_{13}]$$

$$\frac{\mathbf{B}(a,t,\phi) \ \ \mathbf{B}(a,t,\mathbf{O}(a,t,\phi,\chi)) \ \ \mathbf{O}(a,t,\phi,\chi)}{\mathbf{K}(a,t,\mathbf{I}(a,t,\chi))} \ [R_{14}]$$

References

[Allsopp, 2011] Michael E. Allsopp. The Doctrine of Double Effect in US Law: Exploring Neil Gorsuchs Analyses. *The National Catholic Bioethics Quarterly*, 11(1):31–40, 2011.

[Arkoudas and Bringsjord, 2008] Konstantine Arkoudas and Selmer Bringsjord. Toward Formalizing Common-Sense Psychology: An Analysis of the False-Belief Task. In T.-B. Ho and Z.-H. Zhou, editors, *Proceedings of the Tenth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2008)*, number 5351 in Lecture Notes in Artificial Intelligence (LNAI), pages 17–29. Springer-Verlag, 2008.

[Arnold *et al.*, 2017] Thomas Arnold, Daniel Kasenberg, and Matthias Scheutz. Value Alignment or Misalignment–What Will Keep Systems Accountable?, 2017. Presented at the 3rd International Workshop on AI, Ethics and Society at AAAI 2017.

[Bentzen, 2016] Martin Mose Bentzen. The Principle Of Double Effect Applied to Ethical Dilemmas of Social Robots. *Frontiers in Artificial Intelligence and Applications*, 2016.

[Berreby *et al.*, 2015] Fiona Berreby, Gauvain Bourgne, and Jean-Gabriel Ganascia. Modelling Moral Reasoning and Ethical Responsibility with Logic Programming. In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 532–548. Springer, 2015.

[Boolos *et al.*, 2003] George S. Boolos, John P. Burgess, and Richard C. Jeffrey. *Computability and Logic (Fifth Edition)*. Cambridge University Press, Cambridge, UK, 2003.

[Bringsjord and Govindarajulu, 2012] Selmer Bringsjord and Naveen Sundar Govindarajulu. Given the Web, What is Intelligence, Really? *Metaphilosophy*, 43(4):361–532, 2012.

[Bringsjord and Taylor, 2012] Selmer Bringsjord and Joshua Taylor. The Divine-Command Approach to Robot Ethics. In P. Lin, G. Bekey, and K. Abney, editors, *Robot Ethics: The Ethical and Social Implications of Robotics*, pages 85–108. MIT Press, Cambridge, MA, 2012.

[Bringsjord *et al.*, 2014] Selmer Bringsjord, Naveen Sundar Govindarajulu, Daniel Thero, and Mei Si. Akratic Robots and the Computational Logic Thereof. In *Proceedings of ETHICS • 2014 (2014 IEEE Symposium on Ethics in Engineering, Science, and Technology)*, pages 22–29, Chicago, IL, 2014. IEEE Catalog Number: CFP14ETI-POD.

[Bringsjord, 2017] Selmer Bringsjord. A 21st-Century Ethical Hierarchy for Robots and Persons: $\mathcal{EH}$. In *A World with Robots: International Conference on Robot Ethics: ICRE 2015*, volume 84, page 47. Springer, 2017.

[Cushman *et al.*, 2006] Fiery Cushman, Liane Young, and Marc Hauser. The Role of Conscious Reasoning and Intuition in Moral Judgment Testing Three Principles of Harm. *Psychological science*, 17(12):1082–1089, 2006.

[Feldman, 1978] Fred Feldman. *Introductory Ethics*. Prentice-Hall, Englewood Cliffs, NJ, 1978.

[Fletcher, 1998] George P. Fletcher. *Basic Concepts of Criminal Law*. Oxford University Press, 1998.

[Francez and Dyckhoff, 2010] Nissim Francez and Roy Dyckhoff. Proof-theoretic Semantics for a Natural Language Fragment. *Linguistics and Philosophy*, 33:447–477, 2010.

[Gentzen, 1935] Gerhard Gentzen. Investigations into Logical Deduction. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, The Netherlands, 1935. This is an English version of the well-known 1935 German version.

[Hauser *et al.*, 2007] Marc Hauser, Fiery Cushman, Liane Young, R Kang-Xing Jin, and John Mikhail. A Dissociation Between Moral Judgments and Justifications. *Mind & Language*, 22(1):1–21, 2007.

[Huxtable, 2004] Richard Huxtable. Get Out Of Jail Free? The Doctrine Of Double Effect In English Law. *Palliative Medicine*, 18(1):62–68, 2004.

[Kamm, 2007] Frances Myrna Kamm. *Intricate Ethics: Rights, Responsibilities, And Permissible Harm*. Oxford University Press, New York, New York, 2007.

[Khasidashvili *et al.*, 2009] Zurab Khasidashvili, Mahmoud Kinanah, and Andrei Voronkov. Verifying Equivalence of Memories Using a First Order Logic Theorem Prover. In *Formal Methods in Computer-Aided Design, 2009. FMCAD 2009*, pages 128–135. IEEE, 2009.

[Khatchadourian, 1988] Haig Khatchadourian. Is the Principle of Double Effect Morally Acceptable? *International Philosophical Quarterly*, 28(1):21–30, 1988.

[Malle *et al.*, 2015] Bertram F. Malle, Matthias Scheutz, Thomas Arnold, John Voiklis, and Corey Cusimano. Sacrifice One for the Good of Many?: People Apply Different Moral Norms to Human and Robot Agents. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-robot Interaction*, pages 117–124. ACM, 2015.

[McIntyre, 2014] Alison McIntyre. Relevance Logic. In Edward Zalta, editor, *The Standford Encyclopedia of Philosophy*. September, 2014 edition, 2014.

[McNamara, 2010] Paul McNamara. Deontic Logic. In Edward Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 2010. McNamara's (brief) note on a paradox arising from Kant's Law is given in an offshoot of the main entry.

[Mikhail, 2011] John Mikhail. *Elements of Moral Cognition: Rawls' Linguistic Analogy and the Cognitive Science of Moral and Legal Judgment*. Cambridge University Press, Cambridge, UK, 2011. Kindle edition.

[Mueller, 2006] Erik T. Mueller. *Commonsense Reasoning: An Event Calculus Based Approach*. Morgan Kaufmann, San Francisco, CA, 2006. This is the first edition of the book. The second edition was published in 2014.

[Pereira and Saptawijaya, 2016a] Luís Moniz Pereira and Ari Saptawijaya. Counterfactuals, Logic Programming and Agent Morality. In Shahid Rahman and Juan Redmond, editors, *Logic, Argumentation and Reasoning*, pages 85–99. Springer, 2016.

[Pereira and Saptawijaya, 2016b] Luís Moniz Pereira and Ari Saptawijaya. *Programming Machine Ethics*. Springer, Basel, Switzerland, 2016. This book is in Springer's SAPERE series, Vol. 26.

[Pollock, 1976] John Pollock. *Subjunctive Reasoning*. D. Reidel, Dordrecht, Holland & Boston, USA, 1976.

[Rao and Georgeff, 1991] Anand S. Rao and Michael P. Georgeff. Modeling Rational Agents Within a BDI-architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*, pages 473–484, San Mateo, CA, 1991. Morgan Kaufmann.

[Stannett and Németi, 2014] Mike Stannett and István Németi. Using Isabelle/HOL to Verify First-order Relativity Theory. *Journal of Automated Reasoning*, 52(4):361–378, 2014.

[Zalta, 1988] Edward N. Zalta. *Intensional Logic and the Metaphysics of Intentionality*. MIT Press, Cambridge, MA, 1988.