SORBONNE UNIVERSITÉ

École Doctorale 130: Informatique, Télécommunications et Électronique
Laboratoire d'Informatique de Paris 6

THÈSE DE DOCTORAT

nouveau régime

Pour obtenir le grade de Docteur en Informatique

En vue d'une soutenance publique
par

Fiona BERREBY

MODELS OF ETHICAL REASONING

Thèse sous la direction de Gauvain BOURGNE, Jean-Gabriel GANASCIA et Robert VOYER

Devant un jury composé de:

Dr. HDR	Emiliano LORINI	(rapporteur)	Université de Toulouse III
Pr.	Thomas POWERS	(rapporteur)	University of Delaware
Dr.	Virginia DIGNUM	(examinateur)	TU Delft
Pr.	Raja CHATILA	(examinateur)	Sorbonne Université
Pr.	Nicolas MAUDET	(examinateur)	Sorbonne Université
Dr.	Grégory BONNET	(examinateur)	Université de Caen Normandie
Dr.	Gauvain BOURGNE	(directeur)	Sorbonne Université
Pr.	Jean-Gabriel GANASCIA	(directeur)	Sorbonne Université

November 26, 2018

Abstract

This thesis is part of the ANR eThicAa project, which has aimed to define moral autonomous agents, provide a formal representation of ethical conflicts and of their objects (within one artificial moral agent, between an artificial moral agent and the rules of the system it belongs to, between an artificial moral agent and a human operator, between several artificial moral agents), and design explanation algorithms for the human user. The particular focus of the thesis pertains to exploring ethical conflicts within a single agent, as well as designing explanation algorithms.

The work presented here investigates the use of high-level action languages for designing such ethically constrained autonomous agents. It proposes a novel and modular logic-based framework for representing and reasoning over a variety of ethical theories, based on a modified version of the *event calculus* and implemented in Answer Set Programming. The ethical decision-making process is conceived of as a multi-step procedure captured by four types of interdependent models which allow the agent to represent situations, reason over accountability and make ethically informed choices. More precisely, an *action model* enables the agent to appraise its environment and the changes that take place in it, a *causal model* tracks agent responsibility, a *model of the Good* makes a claim about the intrinsic value of goals or events, and a *model of the Right* considers what an agent should do, or is most justified in doing, given the circumstances of its actions. The *causal model* plays a central role here, because it permits identifying some properties that causal relations assume and that determine how, as well as to what extent, we may ascribe ethical responsibility on their basis.

The overarching ambition of the presented research is twofold. First, to allow the systematic representation of an unbounded number of ethical reasoning processes, through a framework that is adaptable and extensible by virtue of its designed hierarchisation and standard syntax. Second, to avoid the pitfall of some works in current computational ethics that too readily embed moral information within computational engines, thereby feeding agents with atomic answers that fail to truly represent underlying dynamics. We aim instead to comprehensively displace the burden of moral reasoning from the programmer to the program itself.

Keywords

Computational Ethics; Answer Set Programming; Event Calculus; Reasoning about Actions and Change; Causality; Counterfactuals.

Résumé

Cette thèse s'inscrit dans le cadre du projet ANR eThicAa, dont les ambitions ont été: de définir ce que sont des agents autonomes éthiques, de produire des représentations formelles des conflits éthiques et de leurs objets (au sein d'un seul agent autonome, entre un agent autonome et le système auquel il appartient, entre un agent autonome et un humain, entre plusieurs agents autonomes) et d'élaborer des algorithmes d'explication pour les utilisateurs humains. L'objet de la thèse plus particulièrement a été d'étudier la modélisation de conflits éthiques au sein d'un seul agent, ainsi que la production d'algorithmes explicatifs.

Ainsi, le travail présenté ici décrit l'utilisation de langages de haut niveau dans la conception d'agents autonomes éthiques. Il propose un cadre logique nouveau et modulaire pour représenter et raisonner sur une variété de théories éthiques, sur la base d'une version modifiée du *calcul des événements*, implémentée en Answer Set Programming. Le processus de prise de décision éthique est conçu comme une procédure en plusieurs étapes, capturée par quatre types de modèles interdépendants qui permettent à l'agent d'évaluer son environnement, de raisonner sur sa responsabilité et de faire des choix éthiquement informés. En particulier, un *modèle d'action* permet à l'agent de représenter des scénarios et les changements qui s'y déroulent, un *modèle causal* piste les conséquences des décisions prises dans les scénarios, rendant possible un raisonnement sur la responsabilité et l'imputabilité des agents, un *modèle du Bien* donne une appréciation de la valeur éthique intrinsèque de finalités ou d'événements, un *modèle du Juste* détermine les décisions acceptables selon des circonstances données. Le *modèle causal* joue ici un rôle central, car il permet d'identifier des propriétés que supposent les relations causales et qui déterminent comment et dans quelle mesure il est possible d'en inférer des attributions de responsabilité.

Notre ambition est double. Tout d'abord, elle est de permettre la représentation systématique d'un nombre illimité de processus de raisonnements éthiques, à travers un cadre adaptable et extensible en vertu de sa hiérarchisation et de sa syntaxe standardisée. Deuxièmement, elle est d'éviter l'écueil de certains travaux d'éthique computationnelle qui directement intègrent l'information morale dans l'engin de raisonnement général sans l'expliciter – alimentant ainsi les agents avec des réponses atomiques qui ne représentent pas la dynamique sous-jacente. Nous visons à déplacer de manière globale le fardeau du raisonnement moral du programmeur vers le programme lui-même.

Mots-clefs

Éthique Computationelle; Answer Set Programming; Calcul des Événements; Raisonnement sur l'Action et le Changement; Causalité; Countrafactualité.

Acknowledgements

I would like to warmly thank my supervisors Gauvain Bourgne, Jean-Gabriel Ganascia and Robert Voyer for their continuous support and invaluable knowledge.

I would also like to thank Alexandre Bazin, Jack Berreby, Grégory Bonnet, Mohamed Amine Boukhaled, Nicolas Cointe, Susan Dunne, Francesca Frontini, Valerie Hanoka, Aurore Marcos, Husein Meghji, Suzanne Mpouli Njanga Seh, Diem Phuong Nguyen, Mathieu Ragueneau, Yassine Souarit, Nadine Taniou, Mihnea Tufis, Bin Yang, and the members of the eThicAa project for their for their precious help, intellectual rigour and kindness in these trying times.

I would also like to express my sincere gratitude the members of the Jury for accepting to review my thesis.

This work is part of the « Ethique et Agents Autonomes » EthicAa project, financed by the French *Agence Nationale pour la Recherche* under the reference ANR-13-CORD-0006.

Contents

List of Figures

List of Tables

Introduction

Context and Motivation

The study of morality from a computational point of view has attracted a growing interest from researchers in artificial intelligence. Indeed, 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, education, financial trading or transportation pose ethical issues that are in this sense particularly pressing, as they may present autonomous agents with decisions that yield immediate or heavy consequences. Computational ethics can also help us better understand morality, and reason more clearly about the ethical concepts that are employed throughout philosophical, legal and technological domains. Confronting ethical theories and philosophical works to the systematicity and logical constraints of programming languages indeed forces us to think about, and make explicit, the underlying mechanisms that characterize those works. It also sheds light on the possible inconsistencies or ambiguities that they may contain. This is what philosopher Daniel Dennett conveys in his statement that « AI makes Philosophy honest » (quoted in [4]). Yet, some works in computational ethics tend to embed moral factors withing their computational engine, without actually generating moral reasoning to speak of. When modelling ethical dilemmas, the moral worth and causal implications of actions as well as the desirability of consequences are atomically afforded, rather than extracted from knowledge of the world and ethical rules. In contrast, our aim here has been to shift the burden of moral reasoning from the user or programmer to the program itself.

For this, we provide a systematic and adaptable framework that enables the agent to appraise both its changing environment and the ethical rules that constrain its actions, so as to determine from these only the correct course of action or an appraisal of the behaviour of other agents. Through a modular architecture, we combine an entirely ethics-free model of the world with an ethical overlayer that the agent can understand and apply back onto its knowledge of the world. What is important to note here is that at the centre of this process lies the notion of *causality*, for only once

the agent can reason about causes and consequences can it fully begin to reason about moral choice and responsibility [22].

Structurally, we present a set of four models: an *action model* enables the agent to represent its environment and the changes that take place in it, a *causal model* tracks the causal powers of actions, enabling reasoning over agent responsibility and accountability, a *model of the Good* makes a claim about the intrinsic value of goals or events and a *model of the Right* considers what an agent should do, or is most justified in doing, given the circumstances of its actions. These are implemented in Answer Set Programming based on a modified version of the event calculus [219]. The resulting framework models ethical reasoning in a rationalised way; such things as emotions or embodied cognition do not interfere. As such, it does not purport to imitate moral reasoning as it is done in reality, but aims to represent moral reasoning as it is prescribed by philosophers: we model *prescriptive* theories, rather than learn or test the *descriptive* ethical behaviour of humans. The structure of the written thesis follows the structure of the framework, with, after introductory sections on philosophical and computational ethics and logic based reasoning, a section dedicated to each model. Before turning to a description of these sections, we make a few short notes on the vocabulary used throughout the monograph.

Disambiguation

Artificial Agent This term pertains to a canonical concept in artificial intelligence which denotes any machine or software which, in a given real or simulated environment, can perform tasks in an autonomous and flexible way towards defined goals [125]. Autonomy in this context can be understood as the capacity to make decisions and perform actions without real time assistance from a human operator or another agent [48]. Here, we often refer to autonomous agents simply as *agents*, while for some discussions also allowing *agents* to denote humans. In many discussed scenarios, human and artificial agents are indeed meant to be commutable, similarly characterised by the fact that they display agency. When specifically discussing artificial agents, we employ *it/its* pronouns, but when the term *agent* pertains to all entities which decide and act, we employ *he/his* pronouns. In addition, we call *pseudo-agents* the characters that populate scenarios and moral dilemmas but that are not the main decision-making agent (even though they may also perform actions in a domain).

Computational Ethics This domain pertains to the production of computational systems for simulating decision-making informed by ethical reasoning. Computational ethics can also be referred to as *machine ethics* or *artificial morality*. But they should not be conflated with *computer ethics* or *cyberethics*, which pertain to the ethical and societal consequences of com-

puters in the information age.

Ethics and Morals Some researchers and research communities distinguish the term *ethics* from the term *morals* (e.g. [50]). However, the distinction made often varies from one work to another. We use these two terms interchangeably, to denote the set of principles that can govern a person's behaviour or the conducting of an activity.

Volitions This term denotes the faculty or power of using one's will. We use it to refer together to actions and omissions, as they both correspond to *a decision made*.

toG/toR In some sections, we use the shorthand expressions *toG* and *toR* to respectively denote theories of the Good and theories of the Right.

Plan

In part I we give an overview of the philosophical field of ethics, looking at normative, applied and meta-ethics. We then discuss the role of dilemmas as a dialectic tool for advancing philosophical inquiry. In the second chapter of this part we review current works in computational ethics, drawing a distinction between top-down and bottom-up models, and presenting a general taxonomy based on framework features such as genericity or implementation.

In part II we review the artificial intelligence domain of *reasoning about actions and change* in which the current work belongs, then describe the fundamental tenants of logic programming and Answer Set Programming in particular. In the third chapter of this section, we present the produced framework in broad strokes, by discussing the four-year and three publications process that led to its final form, summarising its attributes and presenting its modular nature.

In part III we present the *action model*, an adapted version of the *event calculus* which integrates actions, omissions and automatic events in parallel simulations. We discuss agency relative to both actions and omissions and describe our method for scenario generation. We illustrate this contribution through a first proof of concept.

In part IV we present the *causal model*. We begin by characterising our approach to causality with a discussion of counterfactual reasoning and scalar representation. We then present an account of *event-based* causality that is grounded in the architecture of event preconditions and effects, defining four elementary types of causal relations dependent on (a) the nature of the events that compose them (actions, omissions or automatic events) and (b) the direction of the connection (pertaining to produced or avoided outcomes). We study their transitive properties and discuss the specificities of integrating omissions within causal chains. Then, to scrutinise and buttress the causal relations previously identified, we consider four properties of *scenario-based causality*. Inquiring into the other possible versions of modelled scenarios,

we account for simple counterfactual 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?"). We illustrate this work with a second proof of concept.

In part V we present the *model of the Good*, appraising in turn impact-centered and agent-centered theories of the Good, looking at the possible role of rights, values and hedonistic measures for defining it. We follow this with a discussion and a method for assigning weights to events in a domain, then illustrate this through a third proof of concept.

In part VI we present the *model of the Right*, composed of numerous potential principles or *theories* of the Right. We begin by discussing challenges that arise and routes that must be chosen when considering the ethical appraisal of actions, omissions and plans of such actions and omissions. We then model four consequentialist principles that asses ethical permissibility based on the consequences of *volitions*, and study their relationship with principles in decision theory. Next, we model one consequential principle that derives permissibility from the evaluation of generalised *behavioural rules*. In the subsequent sections, we model three deontological principles and investigate one ethical theory based on *causal properties*. We then compare these contributions with related works, illustrating them throughout by means of three proofs of concept. Finally, we present a program that serves to translate the framework's computational output into readable English to facilitate interaction with a human user.

We end this document in part VII, where we conclude and discuss future aims.

The full source code corresponding to the framework, case examples and launching script is available online at: https://github.com/FBerreby/Thesis

Part I

Computational Ethics

Chapter 1

State of the Art: Ethics in Philosophy

1.1 A Concise Map of Ethics

The study of ethics is the study of the beliefs that people may or should have to control their behaviour. A standard tripartite classification splits the field into *normative ethics*, which is concerned with determining, comparing and explaining accounts of the ethically right and wrong [38], *applied ethics*, which is concerned with applying moral rules to particular environments and *meta-ethics*, which is concerned with the status and meaning of ethical concepts. The present work informs applied ethics in that it presents a scheme for designing ethically constrained artificial agents that may act in a variety of applied domains. It also informs normative ethics in that it purports to model the processes that underpin normative ethical decision-making, with the possibility of confronting different views. Finally, it informs meta-ethics in that it examines ethical concepts from an external analytic stance which can help us understand their nature more formally.

1.1.1 Normative Ethics

Normative ethics are concerned with determining the features, principles or characteristics which make agents, actions or outcomes good or bad, right or wrong. They make claims of the form:

Freedom is more important than security.
An action is wrong if it causes undue pain.
A promise should be kept, even if it will make someone unhappy.

Within normative ethics, we distinguish three main avenues.

Consequentialist ethics

Consequentialist theories hinge around the idea that actions are to be morally assessed in terms of their outcome, and can only be right or wrong derivatively, in virtue of what they produce. Accordingly, such theories are sometimes referred to as *teleological* theories, from the Greek words *telos*, "end, purpose" and *logos*, "reason, discourse". A morally right action is one that brings about a good, or the best, state of affairs. Yet, in order to determine the rightness of an action, consequentialists must first establish what constitutes a good state of affairs, i.e. determine what is broadly called "the Good" [2]. This then puts them in a position to assert that actions are morally part of "the Right" as far as they increase the Good. In other words, consequentialism is the view that *whatever notion of the Good* an agent or group of agents adopts, the proper attitude is to promote it. Theories of the Right therefore follow from, rely on and eventually supersede, theories of the Good.

Disagreement among consequentialists about what the Good consists in has nourished a number of strands of consequentialism. It has been said to stem from the happiness or well being of sentient beings (*utilitarianism*), the welfare of others (*ethical altruism*), personal self-interest (*ethical egoism*), net pleasure (*hedonism*), or the respect of individual rights (*utilitarianism of rights*). Disagreement among consequentialists about *how* the Good should be promoted then led to diverging theories of the Right. Harsanyi for instance promoted the utilitarian idea that the overall aggregate Good in a population should be promoted, while Rawls argued for the equitable idea that in existing inegalitarian societies, the Good should be promoted first to benefit those who have the least of it [105, 197]. A further distinction also distinguishes different consequentialist theories of the Right. *Maximising* theories claim that a morally permissible action is one which produces the maximum amount of Good. *Satisficing* theories claim that an action is morally permissible as long as it produces a sufficient amount of Good – maximising is admirable but not necessary in this view. Unlike maximising theories which can be applied directly to a situation, satisficing theories therefore need to contain a clause identifying what counts as sufficient. Attempts at defining the Good will henceforth be referred to as *theories of the Good* or *toG*, and attempts at defining the Right, whether consequentialist or deontological, as *theories of the Right* or *toR*.

Deontological Ethics

Deontological theories (from the Greek *deon*, "duty") hold that the moral value of an action is determined (at least partly) by some intrinsic feature of the action. Usually, this feature is a

rational obligation or prohibition under which the action falls, that constrains the agent to behave in a particular way towards others. For example, a deontological rule may state that lying is unethical, entailing that any utterance which contains a lie is prohibited. Because actions are thought to be right or wrong depending on their conformity with a moral norm or duty, the permissibility of an action is in some ways independent of its consequences. As such, the Right here has priority over the Good: an action may be wrong to the deontologist even if it maximises the Good, and right even if it minimises it. As a matter of example, we can cite Bernard Gert's modern deontological moral system which specifies ten moral rules [84]. These are:

1. Do not kill.
2. Do not cause pain.
3. Do not disable.
4. Do not deprive of freedom.
5. Do not deprive of pleasure.
6. Do not deceive.
7. Keep your promises.
8. Do not cheat.
9. Obey the law.
10. Do your duty.

Deontological theories often face the criticism that they are too demanding, since all actions are seemingly either required or forbidden, and that they are unable to handle situations in which different rules come into conflict. What would Bernard Gert do, for instance, if keeping a promise were to cause pain to someone? Some deontologists, such as Kant, would maintain that certain duties are indeed universal and should be upheld even when they lead to poor results [130]. Others, however, concede that rules are not absolute and can admit exceptions. Gert allows this and states for instance that

> "[...] everyone is always to obey the rule except when an impartial rational person can advocate that violating it be publicly allowed. Anyone who violates the rule when an impartial rational person could not advocate that such a violation may be publicly allowed may be punished."

Another notable characterisation of defeasible moral rules can be found in W.D. Ross's concept of *prima facie duties* [202]. A prima facie duty is thought of as an obligation to which we should adhere except if it is overridden by a stronger obligation, i.e. a duty of more importance. A prima facie duty is therefore more like a strong presumption than an obligation in the classical use of the term. His prima facie duties include fidelity, reparation, gratitude, non-injury, harm-prevention, beneficence, self-improvement and justice. Priority rules are also given, which themselves can admit exceptions. For example, all things equal, non-injury usually overrides all other prima facie duties and fidelity usually overrides beneficence. Likewise, the structure of Asimov's famous fictional Three Laws of Robotics also demonstrates a concern for prioritisation and defeasibility. A law is to be applied to the extent that it does not violate the ones that came before it [13]:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

Virtue ethics

Virtue ethics place their primary emphasis on personal features and motives, such as kindness, bravery or prudence. Such ethics were prominent in ancient ethical theory. Unlike modern philosophers who have tended to focus more on the ethics of actions in context, Athenian philosophers focused on the qualities of a moral person: their reasons for action, their state-of-mind, their relationship to others. Here, the Right is to be understood in terms of what a good person would do. The idea is that morally acceptable choices flow from the cultivation of good character which is itself based in the accomplishment of virtues: they generate habits that dispose us to reason well. This kind of view functions in a similar way to the legal concept of "reasonable person," a hypothetical agent used by courts as a standard against which to measure the typical acceptability of a person's behaviour, for example relative to negligence or gullibility (see for instance *The Man on the Clapham Omnibus* in English law [36]).

But just as consequentialist philosophers diverge in their definition of the Good and deontological philosophers diverge in their proposed sets of duties and obligations, so do virtue ethicists diverge in the virtues they prescribe and the ways in which they classify them. In Plato's Republic, for instance, the study of morality converged around four cardinal virtues: wisdom, courage, temperance, and justice [209]. All other virtues were then thought to derive from these. Working from this, Aristotle proposed additional virtues which he classified as *intellectual* or *moral*. Intellectual virtues can be learned through direct instruction; they include philosophical wisdom (*sophia*), the understanding of scientific knowledge (*episteme*), the rational intuition of necessary truths (*nous*), practical wisdom (*phronesis*) and skill in the arts and techniques (*techne*). Moral virtues, which include temperance, self-discipline, modesty and friendliness among others, are to be learned through experience, i.e. through the practice of living honourably [10]:

> "It is by doing just acts that we become just, by doing temperate acts that we become temperate, by doing brave acts that we become brave. The experience of states confirms this statement for it is by training in good habits that lawmakers make their citizens good."

Good choices lead to strong character which leads to more good choices. It is also possible, following A. Comte-Sponville, to distinguish between *moral* virtues and virtues that are *political*. The former are grounded in a sense of conviction, the latter in a sense of responsibility. For instance, generosity is considered to be a moral virtue in that it essentially demands that, individually, we attempt to be less egotistical than we are naturally. Reversely, solidarity is considered a political virtue in that it demands that, as a group, we attempt to be egotistical "intelligently and together," rather than foolishly against one another [52].

Because ethical theories often receive a monistic treatment in the philosophy literature, consequentialist, deontological and virtue theories are often set up against each other as mutually exclusive. But normative ethicists can adhere to monism or to pluralism. *Moral monism* is the view that one ethical theory matters more than all the others and that all ethical decisions should be made on its basis. *Moral pluralism* is the opposite view that people can switch between ethical theories to make decisions without challenging their moral integrity. The challenge for them resides in determining where and how we want to apply each theory [223]. Within a particular consequentialist strand, theories of the Good can also be monistic or pluralistic. As summarised in table 1.1, we have here presented different strands of normative ethics in a broad manner. The detailed investigation of specific theories will be done in later chapters as we encounter them throughout the thesis. We now turn to a brief discussion of applied ethics.

Table 1.1: Types of normative theories

Type of theory	Emphasise	Promote	Example proponents
Consequentialist ethics	the outcome	the Good	Mill [172]
Deontological ethics	the action	duty	Kant [132]
Virtue ethics	the agent	flourishing	Aristotle [10]

1.1.2 Applied Ethics

Applied ethics are concerned with applying general moral principles to particular situations. They often inform policy making and law. They answer such questions as

> *Is euthanasia ever justified?*
> *Is torture ever just if it results in a long term cease-fire?*
> *In so far as they don't always deter criminals or protect society from criminals, are prisons just a means of vengeance?*

Applied ethics are typically dependent on conceptual commitments to ideas in normative or meta-ethics. For example, if one is a normative consequentialist and a monist, it will be difficult for

them to justify an action in deontological terms even if it occurs within a specific branch of applied ethics. Nevertheless, applied ethics raise new issues by investigating the challenges of applying general theories to specific and real contexts. Various fields make up applied ethics. *Bioethics*, for instance, reflect on issues arising from technical advances in the fields of biology and medicine. An example of applied medical ethics can be found in Beauchamp and Childress's four principles of medical ethics [44]. They are:

> 1. The Principle of Autonomy, which states that a healthcare professional should guarantee the effective exercise of patient autonomy. More precisely, a patient making a decision about their care is considered autonomous if the decision is: (a) intentional, (b) based on sufficient understanding of their medical situation and the likely consequences of undergoing treatment, (c) acceptably free of external constraints (e.g. monetary concerns) and (d) acceptably free of internal constraints (e.g. pain caused by side-effects).
> 2. The Principle of Nonmaleficence, which states that a healthcare professional should never harm a patient.
> 3. The Principle of Beneficence, which states that a healthcare professional should always promote patient welfare.
> 4. The Principle of Justice, which states that healthcare services and burdens should be distributed in a just fashion.

Other types of applied ethics include *business ethics*, which look at practices in corporate settings, *environmental ethics*, which look into the relationship between humans and nature, *cyberethics*, which are concerned with issues arising from the information age. Here, it is important to distinguish cyberethics from computational ethics: the former is an investigation of the ethical consequences of computation, while the latter is a computational investigation of ethics in all possible domains. Though they may be applied in a domain of applied ethics, computational ethics are not, *per se*, applied ethics. However, one branch of cyberethics is particularly pertinent here, the one which looks at the ethical and societal consequences of using and disseminating computers which make ethical decisions. We might call this *the cyberethics of computational ethics*. Indeed, implementing ethics within autonomous agents generates novel ethical hopes, behaviours and problems.

It produces new hopes in that it, for instance, opens the prospect for a morality that is not inhibited by the limiting features of being human, such as having emotions or a bounded rationality. The Stoics believed that moral reasoning should be free of emotional prejudice, and the autonomous agent may well allow for this. It produces new behaviours because, for example, it leads people to change their standards for morality. Malle *et al.* have showed that people make different ethical demands to humans and to autonomous agents [161]. Presented with an ethical dilemma, subjects were more willing to blame an agent from harming someone if the agent was a human than if it was

an autonomous agent. They also preferred autonomous agents to behave in consequentialist ways, while more easily allowing humans not to. This might be interpreted to mean that people expect autonomous agents to make decisions unmediated by such factors as emotion or convention. Finally, implementing ethics within autonomous agents produces new problems because, for instance, the benefit of having humans and computers cooperate in morally charged situations is far from established. In environments where decision making is shared or aided by autonomous agents, there is a risk that they might undermine the moral responsibility of human decision-makers. In a hospital, for example, might a doctor feel pressured to give in to the recommendations made by an artificial system even if she does not agree with them? Might she loose her habit of making moral calls herself as reliance on an external tool becomes pervasive? Might a new-found traceability influence and corrupt decision-making by brandishing the flag of legal liability over what was previously a domain of personal responsibility? Questions of this type underline the timely relevance of Bruno Latour's cautionary claim that, as we interact with and through artifacts, they influence the decisions we make and how we make them [143].

1.1.3 Meta-Ethics

Meta-Ethics are concerned with investigating the meaning of moral judgements (*moral semantics*), the nature of moral judgements (*moral ontology*) and the justification of moral judgements (*moral epistemology*) [78]. They answer such questions as

> *Can we ever know that it is wrong to kill an innocent child?*
> *Are moral judgements an expression of beliefs or of emotions?*
> *Are moral judgements ever true or false?*

Most of us would be more comfortable thinking that the moral judgements we make everyday are grounded in truth or in some reliable rational process. Yet it is not immediately clear how this could be done. Moral truth is not something that we can apprehend through our senses like we apprehend a taste or a sound. It is not something that we can understand or grasp in the way we grasp that all bachelors are men. Nor is it something that we can feel like we feel tired or thirsty. So how might we approach morality? Many philosophical currents tackle this issue, and we summarise a number of them here. We classify them into three categories, but it should be noted that the theories mentioned under one heading may also be derivatively pertinent to another.

Moral semantics

The following theories address the meaning of moral judgements.

Moral realism claims that moral judgements correspond to ordinary facts about the world that are no different from the facts of physical reality described by physics or biology (e.g. [39]). In other words, the meaning of moral judgements is independent from subjective opinion. But moral realists disagree about how we can come to know these facts. *Moral naturalism* claims that we can come to know them through observation only. This is Mill's view [172]. One way to justify this position is to argue that moral facts arise from, or are identical, to natural facts. Just like salt is identical to (or arises from) a combination of sodium and chloride, so is fundamental goodness identical to, say, joy. They are just two ways of looking at the same thing. This view implies that examination of the natural world is relevant to morality. *Moral non-naturalism*, on the other hand, argues that observation is not sufficient and that some moral knowledge is based on intuition or otherwise a priori awareness. This for example is Moore's view [175].

Moral subjectivism claims that moral judgements derive from the conventions or postulates of a group of people or a particular entity. *Divine command theory*, for instance, claims that morality is entirely determined by God, and that moral judgements are to be made in accordance with Him through such means as scripture or religious practice. Adams and Quinn hold this view [1, 193]. *Ideal observer theory*, in a similar but secular fashion, claims that valid moral judgements should be understood as statements that an ideal observer would make in the same situation. The ideal observer is typically held to be neutral, rational, imaginative and well informed. Hume, Firth and Richard Brandt all propose variants of this view [34, 72, 121].

Moral emotivism claims that moral judgements are expressions of our emotions and are not supposed to be true or false. Though moral judgements resemble assertions in their form, this is deceptive. They are closer to such emotional displays as "Good heavens!" or "Yay." Saying that something is wrong is just displaying a felt aversion to that thing. Within this view, moral judgements are devoid of descriptive meaning, and there are no moral facts. Ayer and Russell prone moral emotivism [15, 207].

Moral prescriptivism claims that moral judgements correspond to imperative sentences, such that "Stealing is wrong" equates to "Do not steal." Prescriptivism typically also requires that these statements be universalised in such a way that someone making a moral statement, or injunction, commits to it in all relevant instances. Here, the main function of moral judgements is to recommend behaviours upon others. Moral language, in itself, is uninformative. Hare and Carnap are proponents of this view [42, 102].

Table 1.2 summarises the claims made by these theories and represents the ways in which they converge and diverge.

Table 1.2: Types of meta-ethical semantic theories

Moral Judgements →	Can they be true or false?	What do they express?	Relative to what entity?	How do we acquire them?	Example proponents
Naturalism	yes	propositions	facts	observation	Mill [172]
Non-naturalism	yes	propositions	facts	obser. + intuition	Moore [175]
Ideal observer	yes	propositions	ideal ob.	reason	Firth [72], Hume [121]
Divine command	yes	propositions	God	sacred texts, etc.	Adams [1], Quinn [193]
Emotivism	no	emotions	ourselves	feeling	Ayer [16], Russel [207]
Prescriptivism	no	imperatives	ourselves	introspection	Hare [102], Carnap [42]

Moral Ontology

The following theories address the nature of moral judgements.

Moral universalism claims that moral judgements follow from overarching moral principles that apply to everyone everywhere regardless of where they come from or who they are. What is wrong for me is also wrong for you. Morality is universal by definition: if a moral rule is not universal then it is not a moral rule. Among many others, Chomsky defends a universalist morality [46].

Moral relativism claims that moral judgements are a matter of opinion and are invariably and exclusively dependent on the culture, group or individual from which they are derived. There can be no universal moral truth because there are no objective grounds for favouring one principle over an other. Moral statements can however be true in a non-universal sense. What happens in effect is that people and societies make moral claims based on their specific traditions, practices or beliefs. The moral relativist view is supported by the fact that people tend to make moral judgements in line with the ones that are dominant in their society. Harman takes a moral relativist stance in [103].

Moral nihilism claims that moral judgements cannot exist because nothing has intrinsic moral value. Stealing is neither intrinsically wrong nor intrinsically right. Mackie argues for such nihilism in [159].

Moral Epistemology

The following theories address the justification of moral judgements.

Moral foundationalism claims that we can justify moral beliefs by appealing to other moral beliefs until we reach bedrock beliefs that are "self-evident." Such a view must therefore provide a way of establishing what such bedrocks beliefs are (through a theory of non-inferential

justification) and how to reach other beliefs on their basis (through a theory of inferential justification). Theories of non-inferential justification include:

Moral intuitionism, which claims that intuitions account for our moral knowledge (Sidwick for instance argues for this [220].

Moral rationalism, which claims that moral truths are knowable *a priori*, through reason alone (this is for instance the position of Kant and Plato [130, 209]).

Moral empiricism, which claims that experience and observation yield moral knowledge (this is the position of Hume [121]).

Moral coherentism claims that a moral belief is justified as long as it belongs to a logically coherent and consistent network of beliefs. Though it lacks secure foundation, a body of moral knowledge can still be established by combining the strengths of it components. This is Sayre-McCord's position [213].

Moral scepticism claims that moral judgements cannot be justified since moral knowledge is impossible or unattainable, as claimed in Joyce's work on the "myth" of morality [126].

Beyond the multiplicity of existing theoretical positions, meta-ethical theories do well to be accompanied by non-philosophical theories derived from experimental psychology, neuropsychology, evolutionary biology or linguistics. These fields have identified a multitude of processes, heuristics and biases that constrain moral reasoning. We might cite the concept of bounded-rationality, the finding that when humans reason, the amount of information that is available to us as well as our cognitive capacities are limited, preventing us from judging and deciding optimally. Moral reasoning is no exception to this [85]. Theory of mind, the human capacity of inferring the intentions and states-of-minds of others, enables people to understand and anticipate the actions of others. It has also been shown to participate in moral decision making by enabling feelings of empathy and compassion [41]. Emotions, more generally, regulate moral decision making. People for instance tend to condemn more severely actions that produce negative feelings than actions that do not, even when these are equivalent in moral terms [59]. People make kinder moral decisions if they are pleasantly disposed, as discussed in Ruwen Ogien's aptly titled book, *"Human Kindness and the Smell of Warm Croissants: An Introduction to Ethics*" [183].

Moreover, different meta-ethical theories can present similarities with, entail, or preclude one another. For instance, nihilists are also sceptics. In addition, if meta-ethical theories do not make any normative claims evaluating the moral acceptability of actions or choices, they may have deep implications for the validity, significance and use of such claims. A nihilist cannot hold any deontological principle that claims or implies that *it is itself true*, such as the claim "it is wrong to steal" which implies that it is true that stealing is wrong. Meta-ethical theories provide abstract ways of

thinking about morality, as a result of which they are sometimes referred to as *second-order* moral theories, to be contrasted with *first-order* normative theories. We now turn to the function of moral dilemmas in these theories.

1.2 Dilemmas

1.2.1 The Role of Dilemmas in Normative Ethics

The word *dilemma* comes from the Greek $\delta i \eta \mu \mu \alpha$, which joins δi ("twice") and $\lambda \eta \mu \mu \alpha$, effectively meaning "double proposition". A moral dilemma is a situation in which an agent is faced with a choice between two, or potentially more, possibilities for action that morally come into conflict with one another. They may, for instance, be both morally desirable or both morally undesirable.
Moral dilemmas (and thought experiments more broadly), are purposefully difficult cases that serve to test ethical theories and investigate their capacity for adjustment. In so far as ethical theories are here to help us make decisions, dilemmas serve to build their resilience to the challenges of actually applying them. This means that dilemmas have played a key role in the development of moral philosophy: a common pattern emerges throughout its history whereby a philosopher produces a moral dilemma or thought experiment whose contradictions lead to the formulation of ethical theories by other philosophers to attempt solving them. The reverse also happens whereby a dilemma is put forward as a way of exemplifying how well a particular theory withstands it, lending weight to that theory. These two patterns can occur iteratively and form chains that eventually constitute the philosophical literature on a particular topic. To illustrate this, we will the discuss the well-know trolley problem and its scholarly ramifications.

In [74], Philippa Foot introduced the *trolley problem* to vouch for the doctrine of double effect (DDE). The DDE is, though not exclusively, a Christian doctrine whose first known mention is made in Thomas Aquinas' Summa Theologica. At the time of publication of Foot's paper in 1967, the doctrine did not have a prominent place in moral philosophy, but her work placed it back into the present ethical discourse. We turn to her methodology for argumentation and the original description of the trolley problem:

> "Suppose that a judge or magistrate is faced with rioters demanding that a culprit be found for a certain crime and threatening otherwise to take their own bloody revenge on a particular section of the community. The real culprit being unknown, the judge sees himself as able to prevent the bloodshed only by framing some innocent person and having him executed. Beside this example is placed another in which a pilot whose airplane is about to crash is deciding whether to steer from a more to a less inhabited

> area. To make the parallel as close as possible it may rather be supposed that he is the driver of a runaway tram which he can only steer from one narrow track on to another; five men are working on one track and one man on the other; anyone on the track he enters is bound to be killed. In the case of the riots the mob have five hostages, so that in both examples the exchange is supposed to be one man's life for the lives of five."

This dilemma poses a problem for consequentialists. Indeed, a consequentialist view, such as classical utilitarianism, would have us save the five and kill the one in both cases, so as to maximise the good consequences and minimise the bad ones (assuming we agree that saving lives is good and killing people is bad). Yet intuitively, it seems that we should say "without hesitation, that the driver should steer for the less occupied track, while most of us would be appalled at the idea that the innocent man could be framed" [74]. The dilemma therefore suggests that, at least sometimes, we naturally take into account non-consequentialist moral arguments when making moral judgements (whether we know it or not). It highlights differences in the application of consequentialist and deontological constraints.

Now that a problem has been declared, the author suggests a way to solve it. The doctrine of double effect, which specifically provides a way to handle an action that has both a good effect and a bad effect, is put forward. As described in [136], it goes as follows[1]:

1. *The nature-of-the-act condition.* The act itself must be morally good or at least indifferent.
2. *The right-intention condition.* The agent may not positively will the bad effect but may merely permit it. If he could attain the good effect without the bad effect, he should do so. The bad effect is sometimes said to be indirectly voluntary.
3. *The means-end condition.* The good effect must flow from the action at least as immediately (in the order of causality, though not necessarily in the order of time) as the bad effect. In other words, the good effect must be produced directly by the action, not by the bad effect. Otherwise the agent would be using a bad means to a good end, which is never allowed.
4. *The proportionality condition.* The good effect must be sufficiently desirable to compensate for the allowing of the bad effect.

The doctrine, and in particular the means-end condition, provides a set of deontological criteria which succeeds in explaining the divergence in intuitions between the two cases. It gives a systematic way of understanding where and why we stop favouring consequentialist reasons over deontological ones: in this case, we prefer saving as many people as possible until something about the situation

[1] There are a number of existing variations in the precise formulation of the doctrine, but they are in essence identical.

makes it no longer morally "worth it," here, this something is the fact that we come to *use* a person's death to save others.

But the discussion does not end here. Counter-arguments have been proposed to challenge the claim that the DDE truly accounts for our intuitions. Typically, these arguments have again taken the form of dilemmas or thought experiments, and have materialised into pointed variations on the trolley problem. We will here discuss some such variations, but first we introduce the more commonly used version of the problem which simplifies it (while retaining the same meaning at heart), so that there is no longer talk of planes or mobs but rather trains and tracks.

> (*switch*) A train is running towards five people stationed on train tracks; these people are workmen repairing the track. If the agent does nothing, the train will run over and kill them. However, the agent has the option of actioning a switch that will deviate the train from these tracks and place it onto another set of tracks along which one person is walking. This will kill the person.
> (*push*) There is no switch button, instead there is a bridge above the train tracks on which stands an onlooker. Here, the agent knows that if he pushes the onlooker onto the tracks, the train will run into and kill the onlooker, and stop as a result of the crash, thereby saving the five workmen.

Again, people overwhelmingly tend to morally accept the *switch* action but not the *push* action [92, 110]. But in [237], Judith Jarvis Thomson presented a case where the DDE failed to match our intuitions. The (*loop*) case takes the original (*switch*) case in which the trolley is headed toward five workmen but can be redirected onto side tracks where one innocent bystander stands, but adds that the track loops back toward the five. Therefore, if it were not the case that the trolley would hit the one and stop, it would go around and kill the five. If the agent throws the switch, the bystander on the side tracks dies, if not, the five workmen die (note that if the five were not present, the trolley would not go around and hit the one, but would carry on harmless down the track). It is similar to the original (*switch*) example but differs in that the death of the bystander – if the switch is thrown – prevents the death of the five. According to the DDE, flipping the switch is here impermissible, for it violates the means-end condition. But this is contrary to most people's intuitions [171].

In order to address and explain this discrepancy, Frances Kamm has argued that a distinction is to be drawn between acting *because-of* and acting *in-order-to*. She argues that what she calls the doctrine of triple effect (DTE) can explain the permissibility of redirecting the trolley in this case. The idea is that, beyond the intending/foreseeing distinction of DDE, there is a significant difference between doing something because an effect will occur and doing it in order that it occurs, *whereby doing something because an effect will occur does not imply that one intends that the effect occurs.*

We stop the philosophical discussion of this point here, but we will come back to it in section 9.4.4 where we both model and investigate the claims made within it. Figure 1.1 summarises the analytic process up to here.

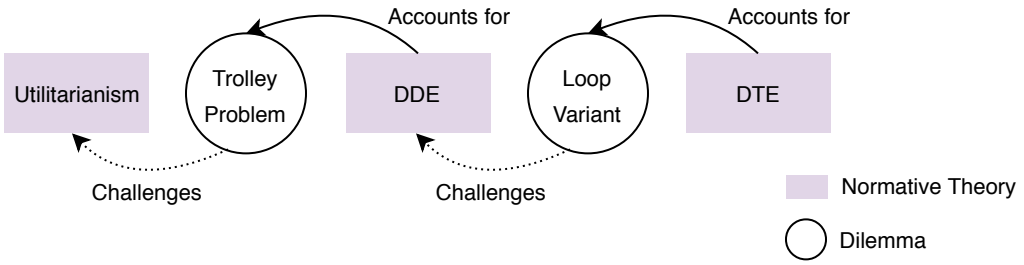

Figure 1.1: From utilitarianism to the doctrine of triple effect

We conclude this section by making a couple of general point about moral dilemmas. It should first be noted that dilemmas may oppose moral reasons of varying nature. They may oppose different values, norms, principles or theories in any configuration. This is that case in Kant's example of *lying to the murderer at the door* from the essay "On A Supposed Right to Lie From Altruistic Motives" [131]. In it, an agent is faced with the choice between lying and telling the truth to a murderer who is asking about the whereabouts of a person he intends to kill. The choice lies between violating the "do not lie" deontological principle or violating the "do not allow harm to come to others" deontological principle. The trolley problem, as discussed, opposes a deontological principle with a consequentialist one. But a moral dilemma may also oppose two instances of the same theory. In the novel Sophie's choice, a mother is given a choice by a Nazi doctor: she must choose to save one of her two children or he will kill them both. Her dilemma, beyond being a personal one, is a moral one which opposes two instances of "do not allow to come to others."

Moreover, a dilemma is not necessarily a dilemma for all. Take Kant's (*lying to the murderer at the door*). Arguably, if we look at this situation from a consequentialist perspective, and deem the death of a person worse than the act of lying, then the conflict effectively falls apart. The case is a dilemma to the absolute deontologist but not to the consequentialist. As such, just as dilemmas are produced in response or as a challenge to ethical theories, so are they more likely to be interesting in relation to these theories than in isolation. Finally, we note that throughout moral philosophy but also in the present work, dilemmas, thought experiments and case examples often depict dismal and gory situations in which death and distress are ubiquitous. Here, this is not exactly out of personal fancy or concern for academic tradition, but because such cases typically, though not uniquely, afford both the writer and reader with situations which are easy to grasp, clearly polarised and challenging enough to give pause. In addition, the challenge of thinking up

new explanatory scenarios has made recurrent the need to find enduring and immutable outcomes for agents and the characters in their stories, so as not to be confounded by endless and mildly productive "what ifs?" The only candidate outcome to have consistently succeeded in fulfilling this need has been, alas, death.

1.2.2 The Role of Dilemmas in Meta-Ethics and Behavioural Psychology

A dilemma is not meant to be realistic, rather, it strips a situation from what is morally irrelevant and threatens to confound our moral intuitions in order to make salient what truly matters. If the reader were to pitch the trolley problem at a social event, without mention of the DDE, it would not be unlikely that they would receive such an answer as "the reason we make different judgements in the two cases comes from the fact that in (*push*) we physically touch and push the man on the bridge, whereas the walker on the tracks in (*switch*) is far removed." This type of intuition can mean two things. For a proponent of the DDE, it may mean that the trolley problem is not sufficiently fine-tuned to present a situation that is completely stripped of extraneous information, and that this information is obscuring the real mechanisms at play in our judgement that are well captured by the DDE. For someone else, it may mean that the DDE does not succeed in explaining our intuitions and that other factors *are* actually at play, for instance factors relating to embodied cognition. One way to test these claims is to fine-tune the scenario (as in [90] for instance). Say, now, that instead of pushing the man of the bridge, the agent can press on a remote button that releases a trap on which the man on the bridge is standing. This makes the man drop onto the tracks, creates a crash and stops the train. Is the *switch* action in this new scenario more morally permissible than the original *push*? If it isn't, then the trolley problem does well to be updated and refined to avoid such confounding factors. But if it is permissible (or *more* permissible) to press on the button to release the trap, as people have held [60, 171], then the DDE is incorrect or insufficient, since it predicts the opposite. Yet in this case, are we ready to admit that such a fickle measure as "degree of contact with the victim" has moral weight? Such a sensorimotor property is not something we would off hand consider to be morally relevant. As Greene puts it in [90], "*Were a friend to call you from a set of trolley tracks seeking moral advice, you would probably not say, 'Well, that depends. Would you have to push the guy, or could you do it with a switch?*'" This precise issue has been investigated experimentally, and researchers found that while subjects typically deemed worse the cases involving physical contact with a victim than those with no physical contact, they then found it difficult to explicitly endorse the distinction as morally valid [93]. How might we explain this discrepancy between moral judgements and their justifications? One way is to appeal to a dual-system theory of moral psychology, in which two separate processes constrain reasoning. A number of such systems have been put forward; they differ in their characterisation of the two

systems but rely on a similar explanatory mechanism.

Rational v. Intuitive Dual-System Theory

Against the cognitive developmentalist assumption that moral judgements are the product of purely conscious effortful reasoning, and in order to account for the role of emotions in moral judgements that has been found through both neuro-scientific [47, 168, 214] and behavioural measures [239, 247], one early influential dual-system opposes *intuitive* to *rational* processes [94, 96, 109][2]. It suggests that moral judgements are at first intuitive, rapid, automatic and emotion laden. It is when people attempt to justify these judgements that they appeal to controlled and rational reasoning processes, potentially leading to an update in judgement [189], or to a certain degree of cognitive dissonance. Occasionally, people *might* reason their way to moral conclusions, but this is not the norm.

This model predicts that the intuitive process will underlie such things as aversion to doing harm to someone by pushing them (because it will be conflated with gut reaction and feelings such as fear or guilt), while the controlled rational process will favour consequentialist reasons favouring the rational aim of saving the most people. Relative to meta-ethical questions on the meaning of moral discourse, some proponents of this account (like Haidt [96]) argue that people use intuitions even when they make consequentialist claims. Others, like Baron or Green [17, 91], commit to the idea that deontological and consequentialist principles have a different ontological status: if consequentialist ones are reached through rational means, deontological ones, in so far as they tend to be produced through the intuitive system, are for the most part post-hoc rationalisations of affect and intuition.

Relative to the trolley problem, this means that the *push* action will more strongly engage the intuitive system and lead people to condemn it, even if they then fail to justify why. Because only the rational system is involved in (*switch*), but both systems are involved in (*push*), this also explains why the latter case presents a more difficult dilemma than the former. However, though it successfully accounts for the above example, it fails to fully explain it: why is it that emotions and intuitions play a bigger role in (*push*) than in (*switch*) or in the trapdoor example? Indeed, Cushman [59] argues that these types of distinctions succeed in descriptive terms but lack explanatory power: "*By analogy, a subway operates automatically while a bicycle requires manual operation; yet, these superficial descriptions fail to explain the actual mechanics of either mode of transportation.*" Moreover, opposing cognition and emotions can be misleading in so far as emotions are involved in both (*switch*) and (*push*) – people die in both cases. Some form of rational information processing is also present in both cases. It seems therefore that a successful

[2]For concision we subsume under one heading different distinctions that can also be treated specifically: intuition/reason, emotion/cognition, automatic/controlled.

dual-system model needs to account for two different ways of processing emotions and rational calculation together, rather than one or the other.

Action-based v. Outcome-based Dual-System Theory

To account for the shortcoming of the above approach, Cushman put forward a different dual-system theory which distinguishes between directly assigning moral weight to actions (e.g. breaking a promise) and choosing actions based on the moral weight assigned to their consequences (e.g. disappointing someone) [59]. This distinction also falls along the lines of the deontological/consequentialist distinction, though not through an appeal to the intuition/reason distinction. It allows that both processes involve elements of emotion and rational cognition.

Motivation for making such a distinction between outcomes and actions can be found in the fact that people have an aversion to *pretend* harmful actions, such as stabbing an experimenter's leg knowing it is protected by PVC casing, or shooting a fake gun [61, 203]. The fact that there is no actual harmful outcome shows that it is not the most salient determinant of the reaction. This type of reaction has also been correlated with moral judgement: personal aversion to performing pretend harmful actions strongly predicted non-consequentialist moral judgements over moral dilemmas [174]. Another piece of supporting evidence comes from the widespread condemnation of victimless crimes, such as consensual incest. People morally condemn such behaviour while at the same type recognising that no harm has been caused [89, 98]. Presumably, then, the moral status of the action is intrinsic to the action and not dependent on its consequences. The action/outcome distinction is also supported by the widely studied feature guiding moral judgement that is the distinction between active and passive harm [18, 61, 64]. People typically consider that it is worse to actively harm someone (e.g. by poisoning them) than to do so passively (e.g. by withholding an antidote). Because the outcome is the same in both cases, there has to be something else that explains the divergence.

In order to explain how people come to make decisions, Cushman draws in [59] a parallel with two types of reinforcement learning methods: model-based and model-free. In a model-based algorithm, an agent weighs different courses of action based on an internal representation of its environment, then chooses the one that has the best outcome. In a model-free algorithm, the agent assesses the value of the actions that are immediately available, for instance by evaluating the rewards it obtains for performing those actions on average. Applied to moral philosophy, a model-free approach means that we base the moral judgement of an action not on its direct outcome but on the history of its past outcomes. Pushing someone will be seen as wrong because it has typically led to harmful outcomes such as making someone cry or being reprimanded for it. Flipping a switch will not, since

that action has not usually led to harmful outcomes. The model-based outcome-sensitive approach will therefore be applied.

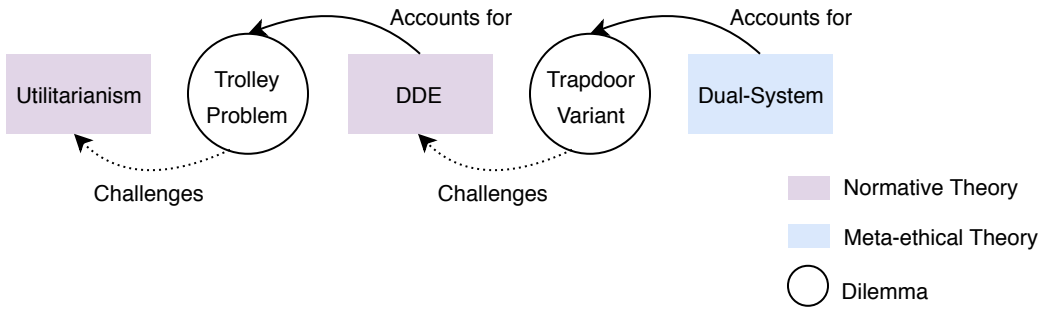

Figure 1.2: From utilitarianism to dual-system theory

It should be noted that while dual-system approaches succeed in accounting for our intuitions, they do not make normative claims. They do not, for instance, say that *push* is indeed worse than *switch*. They simply explain how, as humans, we may come to believe or postulate this. Figure 1.2 represents the analytic process that has led us to here, while table 1.3 summarises the responses to the trolley problem discussed in this section.

Table 1.3: Responses to the trolley problem and its variants

Approach	Theory	*push*	*switch*	*switch in loop*	*switch in trap*	Example justification
Normative Ethics	Utilitarian	✓	✓	✓	✓	*'Maximise utility, understood as the overall well-being of sentient entities.'*
	DDE	•	✓	•	•	*'Do not reach a good effect by means of a bad effect.'*
	DTE	•	✓	✓	•	*'Do not provoke a bad effect in order for a good effect to occur.'*
Meta-Ethics	Dual-System	•	✓	✓	✓	*'Maximise utility, as long as you do not perform actions with a poor track record.'*
Experimental	Intuitions	•	✓	✓	✓	*To be determined.*

✓for permissibility and • for impermissibility

In this section, we have shown the role that dilemmas can play in moral philosophy. They push philosophers to produce novel normative theories to explain and justify moral choices in ambiguous

situations. But they also push philosophers and psychologists to produce meta-ethical and behavioural theories to account for how these moral choices come about and why they are sometimes inconsistent. Some dilemmas, such as the trolley problem, are also so pervasive in the literature that they have effectively become the *lingua franca* of dozens of studies. It should nevertheless be noted that a methodology based on people's intuitions can be flawed. For instance, studies in experimental philosophy and psychology have repeatedly been found to display biases in favour of westerners and males (e.g. [111]). Others have questioned the very endeavour of resorting to fictive moral dilemmas. In [70], Feldmanhall *et al.* for example found that subjects were more willing to harm others in exchange for monetary gain if this occurred in real-life than in an exactly matched hypothetical scenario. Others yet found that people's intuitions about ethical dilemmas vary according to irrelevant factors, such as the context in which the case is being considered (e.g. [150]). We now turn to the following section, where we present a state of the art pertaining to current works in computational ethics.

Chapter 2

State of the Art: Computational Ethics

We here present and compare different approaches made in the field of artificial intelligence towards the production of autonomous moral agents, so as to place and motivate our own approach.

2.1 Characterising Approaches to Computational Ethics

First, a distinction is to be made between two general kinds of approaches, often distinguished on the basis of being *top-down* or *bottom-up* [244].

Top-down approaches infer individual decisions from general rules. A top-down approach takes an ethical theory or principle, apprehends the conceptual and procedural aspects of its application within a computational framework, and applies it to a particular case. This corresponds to the traditional, philosophical, study of ethics whereby individual actions are appraised through the lens of overarching theories, as seen in section 1.1. In a more technical sense, a top-down approach is one which decomposes a task into simpler sub-tasks so as to better organise the efforts necessary to reach an outcome [245]. Top-down approaches to computational ethics are liable to the same pitfalls as top-down approaches to ethics proper: broad ethical theories may be too vague or abstract to be applied successfully in all encountered cases, they may also be subject to debate, be contradictory, or be hostile to unseen situations – and this will show when one attempts to implement them.

Bottom-up approaches infer general rules from individual cases. A bottom-up approach creates an environment in which an agent makes decisions and is rewarded for the ethically acceptable

ones – acceptability being determined through various measures, for instance by comparison to what others have previously deemed acceptable, or through the demonstration of successful interactions with other agents. This trial-and-error feedback loop means that such paradigms can reach high levels of performance in moral decision-making without actually having any general and divisible theory guiding the process. Typically, in the real world, this is how young humans learn morality. Unlike in top-down approaches, which *prescribe* principles of the right and the wrong, these approaches require that such principles be *discovered or built*. They treat normative principles as being implicit components of an agent's behaviour rather than entities dependent on explicit external theories. Proponents of such approaches can even argue that such theories *cannot*, in fact, be made explicit (say for instance because of the potential infinity of exception cases).

A number of works make use of both top-down and bottom-up techniques, and we refer to these as *mixed* models. There are also a number of finer distinctions that variously characterise these works [48]. We may ask:

How generic is the framework? Some models aim to provide an adaptable skeleton from which to infer moral behaviour in specific but variable settings, while others provide an architecture that fits with one setting only. The setting in question may be a domain of application (say a hospital environment) or an ethical view (say utilitarianism). A related question pertains to whether the model is partial or agnostic towards moral judgement. Some models impose an account of morality while others present a canvas upon which users may apply their own. Typically, models that are partial to a theory or set of theories will tend to be specific to a particular setting.

Is the model implemented? Some researchers have proposed theoretical discussions of the tools or challenges for building a given system of ethical decision-making without yet putting them into practice. These participate in the necessary grounding and self-examination of what is a young field, but cannot yield the insights that functional frameworks provide through the actual confrontation of computational (or logical) requirements with ethical concepts.

Is the representation of ethical concepts implicit or explicit? If moral aspects are directly embedded into the decision-making process rather than as stand-alone entities, then we consider that the representation is implicit. Bottom-up approaches will typically have an implicit representation of ethics and top-down models an explicit one. The type of representation will impact the capacity of the model to trace back and explain the moral choice in a way that is understandable to a human. We can also note that there are paradigms that make *some* concepts explicit and others not, by for example explicitly representing moral principles about what an agent is allowed to cause while using an implicit account of causes themselves.

Is there a representation of emotions? Some models propose a purely rational decision-making procedure where others simulate the intervention of emotions with it. The latter allows for some irrational behaviour that can arguably be closer to how humans act in reality, and may for instance enhance human-machine interaction. Indeed, it has been shown that emotion simulation enabled better communication between a system and human users by simulating real feelings of empathy [164]. Rational systems, on the other hand, enable the production of behaviour that is constant, dependable, and free from the pitfalls of human fallibility or error in moral decision making. Such agents may nevertheless have the capacity to take into account the emotions of other agents or users, as part of their environment rather than their internal architecture.

Though we do not delve deeply into them here, a number of works also present broad and conceptual (rather than framework-specific) propositions for developing autonomous agents with ethical reasoning capabilities. In [65], Dignum for instance advocates grounding autonomous AI systems in three founding principles: *accountability*, *responsibility* and *transparency*. She argues that accountability for decisions taken by an artificial agent must emerge from the algorithms and data it makes use of, coupled with a representation of the values and societal norms it abides by. The principle of responsibility emphasises the need for monitoring and disclosing the chain of command which links a given decision with those that are responsible for it, as well as those who incur its consequences. Beyond the agent, this may include users, owners, manufacturers, developers or stakeholders. The third principle underlines the need for transparency and openness relative to the origin and content of used data, as well as to the dynamics which characterise implemented algorithms. She then discusses how different types of frameworks make for variably good candidates in view of respecting these principles. Numerous other works exist in this domain, e.g. [12, 40, 57, 65, 195, 227].

It should be noted that all ethical theories are not equally liable to be successfully modelled. Different normative theories make widely different computational demands. Consequentialist ethics demand that we be capable of reasoning over the outcomes of actions in an environment. Deontological ethics, on the other hand, can be less demanding in so far as they are less concerned with the environment. The tractability of virtue ethics is less clear. Virtues ethics focus on human character, personal motivation. Yet when a virtuous person decides on an action, it is unlikely that they will motivate their choice by explicitly appealing to a virtue. We would not expect them to explain a courageous move by saying that they are courageous. Rather, they may say that they did it because it needed to be done, or because it helped so and so [248]. This kind of example demonstrates the porosity of the boundary between virtue ethics and other kinds of normative ethics. But it also lends weight to the idea that while virtues can provide a valuable and concise way of talking about moral behaviour, they may lack explanatory power. Virtues are dense concepts that subsume nu-

merous moral entities that might be better apprehended separately. For example, trustworthiness might be best understood as a desire to act so as to produce pleasant and reliable relationships (an outcome) or a desire to act in a way that respects others (a duty). The difficulty of modelling virtue ethics is also well illustrated by Socrates's claim that there is just one single virtue, the power of right judgement [190]. Without unpacking what it is, and presumably appealing to things other than virtues, it is impossible to model right judgement. As such, we argue that purely virtue-based ethics are poor candidates for computational modelling in which an explicit representation of concepts is made. They may nevertheless be used in implicit models, such as in bottom-up approaches. We now turn to a state of the art that, while not exhaustive, aims to present the crucial trends and methods that have recently characterised the computational search for the autonomous moral agent.

2.2 Bottom-up Models

2.2.1 Case-based Ethics

A case-based model of ethics works in two steps. First, it aims to learn ethical principles or ethically acceptable behaviour by inferring general rules from sets of case examples, typically produced by experts or by a community [5]. Second, it aims to apply these rules onto new cases to produce autonomous ethical behaviour. Early models of this kind typically succeeded in the first but not the second step. McLaren's SIROCCO, for example, takes as an input a single case of interest and extracts ethically pertinent information by appealing to a preexisting database. This information is then used to inform a human user's ethical choice or assist their rationalisation of it [167]. More recent models have succeeded in reaching definite conclusions autonomously. Anderson *et al.* have developed multiple programs of the sort. *Jeremy* implements Jeremy Bentham's theory of hedonistic act utilitarianism, putting forward a "moral arithmetic" in which one calculates the pleasure and displeasure of those affected by every possible outcome in a situation [7]. "Total Net Pleasure" is calculated by summing the intensity, duration and probability of the outcomes of an action upon all affected individuals, and the chosen action corresponds to the action with the highest value. *W.D.* implements William D. Ross's seven prima facie duties (e.g. beneficence, justice [202]), and appeals to John Rawls's theory of reflective equilibrium to make decisions based on these duties [7, 197]. In these programs, the machine learning procedure generalises principles from numerous cases, then tests the principles on new cases until there have been enough iterations for them to match human intuitions.

More recently, in [181] Noothigattu *et al.* propose a machine learning model of collective decision-making apprehended as a voting procedure. The system proceeds through four steps: human

informed *data collection*, *learning* done through pair-wise comparisons of collected preferences, *summarisation* which concatenates individual models into a single model of collective preferences across all choices, and *aggregation* which deduces from the general aggregate model a collective decision for a given dilemma, i.e. a vote for the best possible alternative. This model of social choice is informed by a novel theory of *swap-dominance* for ranking alternatives and increasing the efficiency of voting rules. It makes use of the data collected through the MIT Moral Machine (http://moralmachine.mit.edu/), which, based on feedback from 1.3 million participants on self-reported preferences, leverages the input of a mass survey to investigate human intuitions regarding decisions made by autonomous vehicles in problematic situations.

Taking another case-based approach, Wu & Lin [251] propose a strategy to ethically constrain the behaviour of a reinforcement learning agent. Their model purports to be low-cost, easy to realise and more faithful to ethical codes than are human decision-makers. It employs the established technique of *reward shaping* in reinforcement learning, which integrates a priori knowledge into the reward function to accelerate the learning process. Adequate policies are generated from human data in the method of [95], which derives a stochastic policy from human feedback by imposing binomial distribution. The ethics shaping function based on these policies then rewards ethically just decisions, punishes unjust ones and remains neutral towards decisions that do not involve ethical considerations. Three experiments show how this work can make the SARSA [206] algorithm perform more ethically. In this work, the separation between the reward function and the ethics shaping function ensures that the burden of codifying ethical decisions is alleviated from reinforcement learning designers.

Casuistry has the advantage of generating a generic framework that can be implemented inclusively within a field of application, but it is often criticised for inheriting the consequences of resorting to reasoning by induction or proximity instead of deduction. Even when their premises are correct, such systems can be unreliable and infer incorrect or problematic conclusions, for example due to under or over learning. Indeed, if the standard for trustworthiness in case-based approaches is that the system performs well many times or most of the time, the fact that they are to be implemented in real life situations with potentially grave consequences can lead one to argue that this requirement is too loose (e.g. [37, 224]). The production of databases containing ethical dilemmas and judgements of these dilemmas is also problematic. It relies heavily on human input, which can be costly or difficult to source, and may be imprecise in so far as it depends directly on the moral intuitions of those building the database. Finally, these systems do not provide an explicit representation of ethical reasoning, meaning that the framework cannot provide a way to reason over different agent's choices. It also means that the justification of an ethical choice will be difficult and that the framework runs the risk of looking obscure and opaque to non-specialists or to those it will come

into contact with.

2.2.2 Text-based Expression Evaluations

This approach aims to learn morality though textual analysis. In this particular work, for instance, Yamamoto & Hagiwara study the co-occurrence of terms bearing a positive or negative moral connotation [252]. The first stage consists in identifying sentences pertaining to morality, such as "Live obediently" and ascribing them a polarity tag indicating whether the sentence refers to something morally acceptable or morally unacceptable. This is given by a positive or negative numerical value. The learning phase then produces a set of analysed expressions, that is, pairs of words to which scores are assigned depending on how many times they occur in positively evaluated sentences minus the number of times they occur in negatively evaluated sentences. This yields a moral valuation of terms. The researchers then investigate the success of the procedure by confronting these results to judgements made by humans.

This approach shares the pitfalls of case-based ethics in that it does not provide a way to explicitly represent, and therefore justify, moral reasoning. Moral information is represented by standalone numerical values that do not contain the trace of their own justification. In addition, the aim here is not for the agent to decide upon a choice but rather to recover the meaning of a text, potentially as part of a wider set of tasks.

2.3 Mixed Approaches

A number of works appeal to both bottom-up and top-down methods. Anderson, Anderson and Armen's *MedEthEx*, for instance, implements Beauchamp's and Childress' Principles of Biomedical Ethics to advise healthcare workers faced with ethical dilemmas [6]. It presents a mixed approach because even though it learns from cases, these cases are integrated into the learning algorithm as high-level descriptions using top-down concepts denoting principles to respect. Using logic programming and inductive concept learning, the same authors proposed *GenEth*, a general ethical dilemma analyser which serves to codify the ethical principles that pervade a given domain [3]. The proposed codifying is done according to the following representation scheme:

Features represent the ethically relevant aspects of an action, such as whether it causes harm or respects an other agent's autonomy.

Duties are values which denote the extent and polarity of the duty an agent has towards either maximising or minimising a given feature. Typically, the better a feature (e.g. respect), the more strongly an agent has a duty to maximise it and the worse the feature (e.g. physical harm), the more strongly an agent has to minimise it.

Actions are apprehended through a purely ethical prism. Each action is represented as a tuple of integers which represent the extent to which it affects given duties.

Cases compare pairs of actions.

Principles represent preferences across actions.

As a matter of example, consider a situation in which a driver has been driving recklessly in and out of his lane for no justifiable reason, and where the aim is to appraise whether the automated control of the car should take over. Five features are considered: 1) prevention of collision, 2) staying in lane, 3) respect for driver autonomy, 4) keeping within speed limit and 5) prevention of immanent harm to persons. Given this set of features, the *take control* action chosen by the automated system corresponds to the duty values (1, 1, –1, 0, 0), and can be compared to the duty values corresponding the the *do not take control* action (1, –1, 1, 0, 0), yielding through subtraction a case result of (0, 2, –2, 0, 0). From sets of such cases, the framework then generates general characterising principles, and tests their adequacy through an adapted Turing test.

Likewise, Dehgani *et al.*'s MoralDM [62] introduces a mixed model of ethical decision-making informed by psychological research findings on both utilitarian and deontological modes of reasoning. It takes into account contextual aspects and functions across two types of mechanisms: *first-principles reasoning*, which bases decisions on culturally established rules such as sacred values, and *analogical reasoning*, which compares considered cases with cases solved through past decisions. This second mechanism makes use of the Structure-Mapping Engine [69], a computational model of similarity and analogy based on Gentner's 1983 structure mapping theory of analogy in humans. Moral dilemmas are represented using predicate calculus and the model is implemented using the FIRE reasoning engine and its underlying knowledge base, which provides formal representations about everyday objects, people, events and relationships. The model is tested on two psychology experiments. Later, in [31], the efficiency of analogical generalisation in MoralDM was improved with the addition of structure mapping.

More recently, Conitzer *et al.* [53] discuss two promising paradigms for designing moral decision-making frameworks which avoid appealing to ad-hoc rules. The first proposal consists in a game-theoric approach for representing moral dilemmas. It consists in a generalisation of game trees or *extensive form*, supplemented with mechanisms that enable the agent to appraise actions beyond pure game-theoric utility. The second proposal suggests appealing to machine learning techniques and discusses ways to assemble an effective training set of labelled moral decision cases. To achieve this, the authors for instance suggest resorting to psychological frameworks of moral foundation such as Haidt and Joseph's five moral foundation [97]. Finally, they suggest combining these two approaches to expand the capacities of each one.

Mixed frameworks have the advantage of combining the computing capabilities of very different mechanisms, producing powerful and expressive tools. This can for example allow coupling agent learning with the explicit representation of ethical concepts, working towards a rich agent architecture. However, they also inherit some of the limitations of both approaches, such as the difficulty of resorting to human-made atomic input and lack of unequivocal reliability relative to bottom-up processes, or high computational cost relative to top-down ones.

2.4 Top-down Models

2.4.1 Belief-Desire-Intention Architectures

A number of works have taken to updating the classical Belief-Desire-Intention (BDI) model to include moral constraints. The BDI architecture [35, 196] proposes to organise the reasoning of a rational autonomous agent by distinguishing the following elements:

- a set the agent's *beliefs*, which is the mental representation of information it has on the state of the world;
- a set of the agent's *desires*, which determines the goals to be achieved;
- a set of the agent's *intentions*, which contains the actions that the agent will perform to achieve its goals.

A BDI agent is typically a rational agent who acts so as to satisfy its desires. The decision-making process generally sees the agent produce a set of plans (sequences of actions) to be applied in situations (which correspond to sets of beliefs), with the view of satisfying its desires. It then apprehends changes in the situation resulting from its action (or that of others) through mechanisms of perception and communication that update its beliefs and desires.

In [238], Tufis & Ganascia propose the following extension paradigm to integrate moral norms. To begin, the agent's mental states are initialised. It then observes its environment and detects emerging norms, which are transmitted as messages from senders that the agent considers to be trusted normative authorities. This allows the agent to acquire new abstract norms and store them in a database which also serves to delete the obsolete ones. Here, the authors appeal to the definition of an abstract norm given in [56] where an abstract norm is a tuple:

> $n_a = \langle M, A, E, C, S, R \rangle$, where M is the modality (prohibition F, permission P or obligation O), A and E are the activation/expiry conditions, C is the logical formula to which M refers, while S and R are the sanction/reward for breaking/respecting the norm.

In order to decide whether to adhere to a norm, the agent confronts it to its own intentions and selects it if it is consistent with them. The internalisation of a norm then leads to possible updates in the agent's set of desires, inducing possible further updates of all mental states in the following iterations of the execution loop.

In [20], Battaglio *et al.* take a similar approach to which they add an emotional dimension. The authors see emotions as being central to moral decision making, arguing that they play a role both in enabling an agent to become aware of moral precepts and issues, and in permitting it to justify its resulting choices. As such, within the proposed architecture, the plans formulated by an agent to satisfy its desires and values also take into account the emotions that they might provoke. Changes in the agent's environment can also impact on its emotional state. The introduction of such aspects is seen as way of producing feelings of empathy, either simulated between artificial agents, or real, with human users.

2.4.2 Multi-Agent Simulations

In [49, 50], Cointe *et al.* combine an extension of the BDI framework and first-order logic to reason over ethical behaviour in a set of cooperating agents within simulated multi-agent environments. Their model is designed as a module that can be plugged into existing architectures to provide an ethical layer within existing decision processes in cooperation. In the first stage, the agent receives as an input a set of knowledge about moral values and rules, as well as relations of preference between these principles. The agent then evaluates the actions that are possible and those that are desirable, and confronts them to its set of moral values and rules so as to provide a set of morally evaluated actions. These are then assessed *in context*, to determine whether they indeed should be performed in a particular configuration, for example by weighing the degree of desirability of a given action against its degree of morality. This decisional process is then used to enable the judgement by the agent of other agents present in the environment. By replacing its own beliefs, desires and moral precepts by that of another agent, the first agent can reconstruct, partially, or entirely, the path that led the second agent to reach a decision. This allows the authors to identify and model some properties that characterise moral judgements within groups of agents. For instance, they distinguish three degrees of moral judgement:

- *Blind ethical judgement*, whereby the judging agent has no information about the internal states of the judged agent, but only knows of its behaviour.
- *Partially informed ethical judgement*, whereby the judging agent has partial information about the internal states of the judged agent in addition to full knowledge of its behaviour. The judging agent may variously be aware of the other agents' beliefs, desires, or moral precepts.

- *Fully informed ethical judgement*, whereby the judging agent has complete knowledge of the behaviour and internal states of the judged agent.

The capacity for agents to judge their own and others' behaviour is a necessary part of allowing them to act collectively and in conformity with given ethical constraints, and this line of work provides a stepping stone towards the formulation of collective ethics.

In [216], Serramia *et al.* provide a multi-agent framework in which the moral values associated to certain norms can be taken into account as an additional decision criterion beyond those regarding norm representation and associated costs. Taking as an input the collection of candidate norms given by a regulatory authority and the moral values shared by a society, the model encodes the decision-making problem as a linear program which can then be solved by solvers such as the CPLEX LP solver. In terms of application, the authors argue for its use as a decision-support system for citizen driven law-making.

2.4.3 C-P Nets

In [155], Loreggia *et al.* propose an approach based on CP-nets, graphical structures which capture conditional independence assertions. They suggest a way to evaluate whether subjective ethical preferences are compatible with exogenous principles such as ethical principles, feasibility constraints, or safety regulations. They also model a notion of *distance* between CP-nets that helps to determine whether agents are *close enough* to given principles, enabling them to find a balance between their own preferences and exogenous ethical requirements.

2.4.4 Logic-based Frameworks

Philosophers typically produce work in which ethical theories and case examples are formulated in declarative form, which they reason over using formal and informal logic. The choice of appealing to logic-based architectures for building autonomous moral agents is therefore an intuitive one. It is also our choice. We first discuss deontic logic, then move on towards action-centred logics which more closely resemble our own work.

Deontic Logic

Deontic logic is a branch of symbolic logic that reasons about *what ought to be* by adding to elementary logic special ethically charged operators. In Standard Deontic Logic, the operator $\bigcirc$P stands for *it ought to be the case that P*, where P is a proposition or state of affairs [43, 112]. This operator is then manipulated through two rules of inference:

$$\textbf{Obligation} \qquad \frac{\mathrm{P}}{\bigcirc \mathrm{P}}$$

meaning that if P is proved, then it ought to be the case that P, and

$$\textbf{Modus Ponens} \qquad \frac{\mathrm{P},\ \mathrm{P} \rightarrow \mathrm{Q}}{\mathrm{Q}}$$

Standard Deontic Logic also relies on three axiom schemas:

A1 All tautologous well-formed formulas.
A2 $\bigcirc(\mathrm{P} \rightarrow \mathrm{Q}) \rightarrow (\bigcirc \mathrm{P} \rightarrow \bigcirc \mathrm{Q})$
A3 $\bigcirc \mathrm{P} \rightarrow \neg \bigcirc \neg \mathrm{P}$

Deontic logics have the advantage that well established theorem-proving methods can be applied to their results. In so far as trust is a great concern in the fields of application of computational ethics, this is advantageous. As a matter of example, researchers from the Rennsselaer Polytechnic Institute modelled a straightforward utilitarian deontic logic that they buttressed through theorem proving software to generate formal proofs about the adequacy of different ethical codes [37]. Though this was promising, it encountered a number of shortcomings, such as the inability for the agent to reason about its behaviour in the case that it failed to meet an obligation. Another paradigm appealing to deontic logic is Arkin's Ethical Governor which aims to moderate the use of lethal force by armed military drones in the battlefield by representing the Laws of War and Rules of Engagement [11]. This particular paradigm is domain specific, and only a small subset of ethical rules may be modified to be adapted to the specific military mission that is undertaken.

In [191], Powers argues for a mixed approach informed by rule-based deontic logic, drawing attention to its virtue of enabling computational tractability. He then demonstrates how Kant's categorical imperative lends itself well to this type of implementation, and investigates the challenges of such an enterprise. For this, he discusses three broad approaches for developing a Kantian deontological agent and investigates their ramifications. First, he introduces the idea that a Kantian formalism can be used to determine formulas for a deontic logic system. Given Az an action performed by z, Cx a circumstance x and Py a purpose y, this kind of system purports to produce output of the the following sort:

$$\forall z \exists x \exists y (Cx \wedge Py) \rightarrow Az(A \textit{ is obligatory for } z)$$
$$\forall z \exists x \exists y (Cx \wedge Py) \rightarrow \neg Az(A \textit{ is forbidden for } z)$$
$$\neg \forall z \exists x \exists y (Cx \wedge Py) \rightarrow Az \textit{ and } \neg \forall z \exists x \exists y (Cx \wedge Py) \rightarrow \neg Az(A \textit{ is permissible for } z)$$

But, Powers argues, this type of output can only be meaningful and non-trivial if it can be probed for self-contradiction as well as tested for consistency with background theories such as normative commonsense rules. He then defends the idea that non-monotonic logics are more adapted to this kind of reasoning than are monotonic ones. Finally, he discusses a logical view of ethical deliberation akin to belief revision in which an overarching ethical theory would be built from a database of coherent maxims and their consequences. Though not based on case examples, this part of the proposition can be viewed as a bottom-up endeavour in that it infers from individual maxims a coherent whole.

STIT Logic

Standard Deontic Logic has come up against a number of difficulties on the basis that it has no way of reasoning about agents and formalising the notion that their *actions* may be obligatory, even while this is central to the possibility of implementing such logics in embedded systems. Efforts to relieve this issue have resulted in a number of formalisms often coined as "AI friendly" deontic logics. One such formalism is Horty's utilitarian formulation of multi-agent deontic logic axiomatised in [88]. Its essential contribution is the assimilation of deontic operators with indeterministic temporal logic, ultimately leading to deliberative STIT logic. With roots in formal philosophy, STIT logic is best understood when cast against the background of the theory of branching time, itself originally developed by Prior and enhanced notably by Thomason [119, 192, 235]. Horty best describes it as follows [119],

> The theory is based on a picture of moments as ordered into a treelike structure, with forward branching representing the openness or indeterminacy of the future, and the absence of backward branching representing the determinacy of the past. Such a picture leads, formally, to a notion of branching temporal frames as structures of the form $\langle Tree, < \rangle$, in which *Tree* is a nonempty set of moments and $<$ is a treelike ordering of these moments – an ordering such that, for any m_1, m_2, and m_3 in *Tree*, if $m_1 < m_3$ and $m_2 < m_3$, then either $m_1 = m_2$ or $m_1 < m_2$ or $m_2 < m_1$. A maximal set of linearly ordered moments from *Tree* is a *history*, representing some complete temporal evolution of the world. If m is a moment and h is a history, then the statement that $m \in h$ can be taken to mean that m occurs at some point in the course of the history h. Of course, because of indeterminism, a single moment might be contained in several distinct histories: we let $H_{(m)} = \{h\colon\ m \in h\}$ represent the set of histories passing through m, those histories in which m occurs.

With roots in [23], STIT logic then adds, and takes its name from, the concept of *seeing to it*

that. This feature is captured by the so-called "stit" modal operator, which denotes that an agent (or coalition of agents) sees to it that ψ iff ψ is true in all histories compatible with a choice (resp. combination of choices) chosen by the agent (resp. coalition of agents). As such, STIT logic semantically functions around the notion of agent choices which lead to aimed-for states of affairs. But these choices, which can effectively be understood as actions performed by agents, do not have any explicit label, so that they are represented in a purely extensional way as a set of states: acting is akin to picking a set of such states over others. Based on this, Horty proposed in [118] an extension which saw the creation of additional operators capturing notions such as the fact that an agent *ought to see to it that* ψ.

In more recent years, the study of STIT logic has shifted in the direction of theoretical computer science, in particular serving as a tool for the logical foundation of multi-agent systems as well as game theory. In this vein, Lorini [156] proposed a rationalist and epistemic model of morality based on DL-MA (Dynamic Logic of Mental attitudes and joint Actions), an extension of STIT logic that also contains an explicit representation of actions. In this system, every agent performs one action per time point and the actions of all agents occur in parallel. The paper then aims to identify different sources for agent motivation and investigates possible conflicts between desires and moral values. Within the multi-agent system, the architecture of individual agents is defined in such a way that each agent's appraisal of a history (sequences of joint actions) is a function of its *degree of moral sensitivity*, that is, its inclination towards more personal desire-satisfaction or more moral ideal-satisfaction. Each agent can fall somewhere on the spectrum between *purely self-regarding agents* who act so as to maximise the satisfaction of their desires, and *purely moral agents* who act so as to maximise the fulfilment of their moral values. In [157], Lorini *et al.* presented another extension of STIT logic aimed towards the logical analysis of responsibility attribution in individual and collective moral settings. The authors add three new types of knowledge and common knowledge modal operators contingent on the time of choice: an *ex ante* operator pertaining to the agent's knowledge before it makes a choice, an *interim* operator pertaining to the agent's knowledge after it makes a choice but prior to knowing the choices of other agents, and an *ext post* operator pertaining to the agent's knowledge after it makes a choice and after the choices of others are made available. This work also investigates the role of emotions relative to these epistemic states, particularly relating to pride, guilt, moral approval and moral disapproval. We will return to this line of work pertaining to responsibility attribution in the discussion of our *causal model* in section 7.4.

We will also discuss in more detail the relation between our framework and branching time theory when investigating certain consequentialist theories of the right, starting in section 9.2.1.

Beyond extending existing temporal logical frameworks to include explicit representations of actions,

a dedicated area of research in logic-based artificial intelligence focuses on this more directly, known as *Reasoning about actions and change*. The event calculus, which forms the basis for the present framework, falls within this area. For this reason, we present it in more detail in a specific section (section 3). To the best of our knowledge, there have not been any other attempts at modelling ethical reasoning and decision-making through a logical language for reasoning about actions and change.

Logic Programming

Instead of appealing to philosophical and formal logic, some researchers make use of the expressiveness and tools directly supplied by logic programming systems. Resorting like the present work to Answer Set Programming, Ganascia modelled a number of ethical rules by drawing on the non-monotonic properties of this formalism. Modelled principles include the principle of least bad consequence, Kant's categorical imperative or Benjamin Constant's moral particularism [75–77]. Benjamin Constant held the position that morality rests on multiple defeasible principles that are more or less general and contingent on the situations in which they are to be applied. Unlike Kant who held that lying is always wrong, Constant for example argued that it can be morally acceptable to lie if the person being lied to does not deserve the truth. This, argues Ganascia, implies that where Kant understands lying as a public speech act which defers to a single transmitter agent, Constant understands lying as a communication act which defers to both a transmitter agent and a receiver agent. This means that the corresponding rules for representing Constant's claims must contain speech act predicates in which an argument denotes the receiver. Considering Kant's *lying to the murderer at the door* case, Constant's argument would be that the felon does not deserve the truth. Ganascia models this in the following way.

```
act(P,A):-action(A),not unjust(A).
unjust(lie(P,PP)):-person(P),proposition(PP),not non_deserve(P,PP).
unjust(A):-consequences(A,murder).
non_deserve(P,PP):-person(P),proposition(PP),consequences(know(P,PP),murder).
```

In a similar way, Pereira and Saptawijaya proposed appealing to prospective logic programming to allow an agent to look ahead at the consequences of hypothetical moral judgements [185]. Moral reasoning is here modelled via a priori constraints and a posteriori preferences on abductive stable models. Agents are equipped with a knowledge base and an ethical theory from which they prospect abductive extensions that also fit with their goals. At the beginning of each cycle, the agent selects, among a set of goals bound by integrity constraints, which ones to aim for. Next, the agent determines which are relevant abductive hypotheses through the application of contextual preference rules, and applies forward reasoning to infer a set of consequences. These can then be

used to determine a posteriori preferences with relation to an ethical theory. If needed to enact these preferences, the agent can acquire additional information through external oracles. In this case, a second round of prospection takes place in order to account for the ramification of the new information.

These works valuably point to the expressive power of non-monotonic and abductive properties found in logic programming for modelling ethical reasoning. However, they do not in their current state present integral agent architectures, nor do they represent situational and ethical factors independently. The present work is an attempt at advancing in that direction. For this reason, we will return to these works in section 9.6 on related works. To conclude, we compile a taxonomy of the most prominent works presented in this state of the art in table 2.1, summarising their relation to the properties discussed in the introduction relative to the type of approach, its genericity, its operational implementation, the explicit representation of ethical concepts and the appraisal of emotions.

Table 2.1: Computational ethics: State of the art taxonomy

Category	Work	Type	Formal approach	Genericity	Implemented	Explicit	Integrates emotions
Case representation language	McLaren, 2006 [167]	↑	Extract-Transform-Load, analog retrieval	*ds*: enginering, *ts*: NSPE code	✓	✓	•
Learning morality from text analysis	Yamamoto *et al.*, 2014 [252]	↑	natural language processing	general	✓	•	✓
Investigation of dilemmas	Anderson *et al.*, 2014 [3]	↑↓	logic programming, inductive concept learning	general	✓	✓	•
Collective decision frameworks	Lorini, 2012 [156]	↓	DL-MA logic	general	✓	✓	•
	Lorini *et al.*, 2013 [157]	↓	STIT logic	general	✓	✓	✓
	Noothigattu *et al.*, 2017 [181]	↑	machine learning	general	✓	•	•
	Cointe *et al.*, 2016 [49]	↓	BDI, logic programming	general	✓	✓	•
	Serramia *et al.*, 2018 [216]	↓	norms, linear programming	general	✓	✓	•
Individual decision frameworks	Wu & Lin, 2017 [251]	↑	reinforcement learning	general	✓	•	•
	Ganascia, 2007 [76]	↓	logic programming	*ts*: Kant,Constant	✓	✓	•
	Pereira *et al.*, 2007 [186]	↓	prospective logic programming	*ts*: DDE	✓	✓	•
	Arkin *et al.*, 2009 [11]	↓	deontic logic	*ds*: warfare, *ts*:Laws of War	✓	•	•
	Battaglino *et al.*, 2013 [20]	↓	BDI model, OOC emotion theory	general	✓	✓	✓
	Tufis *et al.*, 2015 [238]	↓	BDI model	general	✓	✓	•
	Bringsjord *et al.*, 2006 [37]	↓	deontic logic	general	✓	✓	•
	Pereira *et al.*, 2017 [187]	↓	logic programming	general	•	✓	•
	Loreggia *et al.*, 2018 [155]	↓	C-P nets	general	✓	✓	•
	Powers, 2006 [191]	↑↓	non-monotonic deontic logic, belief revision	*ts*: Kant	•	✓	•
	Dehghani *et al.*, 2008 [62]	↑↓	analogical reasoning	general	✓	✓	•
	Conitzer *et al.*, 2017 [53]	↑↓	game theory, machine learning	general	•	•	•
	Anderson *et al.*, 2006 [6]	↑↓	machine learning, inductive logic programming	*ds*: healthcare, *ts*: Beauchamp & Childress	✓	✓	•

The *explicit* header denotes whether the model provides an explicit representation of ethical concepts. ↑ for bottom-up approaches, ↓ for top-down approaches, ↑↓ for mixed approaches; *ts* for theory specific frameworks, *ds* for domain specific frameworks; ✓for yes, • for no

Part II

Logic-Based Reasoning

Chapter 3

State of the Art: Reasoning about Actions and Change

The study of *Reasoning about actions and change* is an important research area within artificial intelligence, for these are fundamental aspects of the worlds in which intelligent agents evolve. Many types of formalisms have been proposed to attempt modelling them, including the event calculus [63, 142, 173, 178, 217–219], the situation calculus [145, 199], the fluent calculus [114, 234], features and fluents [210, 211], temporal action logics [66, 179] and action languages (including STRIPS, PDDL, Language A, Language B, Language C, AQL, Language R) [71, 80, 82]. These frameworks are widely used within the domains of robotics and artificial intelligence, in particular for tasks that necessitate automated planning. Though they vary in multiple ways, have often been developed in relative isolation and present formal relationships that are not always clear, they typically share a number of semantic features [80, 151, 179]. Fundamentally, they all describe the way in which actions affect and transform a situation, that is, affect the states of a system over time, and where an agent's knowledge about a domain is represented by declarative action descriptions that consist in a set of logical formulas. They hinge around the notion of *transition systems*, which themselves operate from *action signatures*. An archetypal action signature consists of three nonempty sets: a set of value names, a set of fluent names and a set of action names. *Fluents*, which are time-varying properties of the world, have a specific value in a particular state of the world (such as a point in time, or a particular stage in a defined sequence). *Actions*, or in some cases the combination of the effects of simultaneous actions, are executed in some state of the world and affect it by transforming the value of its fluents, leading to a resulting state and enabling the situation progress iteratively. In some cases, the resulting state is not uniquely determined by the impact of the action and the

initial state. Accordingly, an action is called *executable* and *nondeterministic* if it can result in at least one new state (from the original state of the world), and called *deterministic* if it can result in at most one new state. In addition, if an action either makes a fluent true or false, we say that it has a *propositional signature*. Other values, which are not necessarily binary (beyond truth and falsity), may also be assigned to fluents. The cumulative effect of a sequence of actions is known as a *trace*, and this global dynamic is captured by *state transition systems* [80]. Another way of thinking about this is by comparing transition systems to directed graphs whose nodes correspond to states, and whose edges correspond to transitions describing the action occurrences [67]. We now turn to introducing the event calculus, and discuss its relation to two closely related frameworks, the situation calculus and the fluent calculus. This then enables us to motivate our choice of using it.

3.1 The Event Calculus

The event calculus was first introduced in a 1986 paper by Bob Kowalski and Marek Sergot [142] to represent and reason about the effects of actions based on points in time rather than on situations, as in the previously existing situation calculus [178]. First employed in database applications, it has since then been reformulated to fit other forms of logic programming, classical logic and modal logic, and used in wider contexts such as planning, abductive reasoning and cognitive robotics [146, 173].

The event calculus infers what is true given what happened when and the impact of particular actions. It combines a narrative of events with the effects of actions. For instance, given that running makes Adam tired, and that Adam runs at 8:00, the event calculus extracts the conclusion that Adam is tired after 8:00. The event calculus can perform:

Deductive tasks In this case *the narrative of events* and *the effects of actions* are given, as in the example, and we seek to find out what is true at what time. Deductive tasks include temporal projection or prediction, where we seek to find out the outcome of a known sequence of actions.

Abductive tasks In this case *what is true at what time* and *the effects of actions* are given, and we seek to find out what is the narrative of events that led to this. Abductive tasks include explanation, postdiction, diagnosis and planning.

Inductive tasks In this case *what is true at what time* and *the narrative of events* are given, and we seek to find the effects of actions. This means we seek to establish a set of general rules that accounts for the observed data. Inductive tasks include theory formation, scientific discovery and learning.

3.1.1 Features of the Event Calculus.

A set of objects (e.g. `gun`), actions (e.g. `shoot`), situation-independent predicates (e.g. `agent(adam)` which denotes that adam is an agent), and fluents (e.g. `has(adam,gun)` which denotes the state that adam has a gun) characterise the basis of the event calculus. They define the domain and enable the creation of predicates that let the situation evolve, constrained by the following three types of axioms: *initial database axioms* enable the description of the initial situation of the simulation, *event precondition axioms* describe the conditions that must hold in order for an event to occur at a given time, *event effect axioms* specify all the ways the value of a particular fluent can be changed by the occurrence of an event. These features participate in allowing, among others, the following expressive possibilities:

Actions with indirect effects Actions might have effects beyond those described explicitly by their associated effect axioms. Although it is possible to encode these indirect effects as direct ones, the use of constraints that describe indirect effects ensures a more efficient modular representation. For example, state constraints describe the logical relationships that hold between different sets of fluents at all times, therefore, if an action impacts a fluent, it will also indirectly impact the fluents that are related to it.

Actions with non-deterministic effects Actions with non-deterministic effects may in the event calculus be handled via what are called "determining fluents". The role of these (non-inertial) fluents is to determine whether or not an event might initiate or terminate another fluent. For example, given some agents playing at the Russian roulette, the event `shoot` non-deterministically may or may not result in the firing of the bullet. The outcome will depend on the status of a determining fluent such as `bulletPresent`. If it holds, then the `shoot` action will initiate the `bulletFired` fluent, and not so if it does not hold. Determining fluents are not themselves initiated or terminated by event occurrences.

Compound actions Actions may be composed of other actions, as is often useful in hierarchical planning [219]. For example, the atomic actions `setTable` and `cookFood` might together correspond to the compound action `prepareLunch`. The challenge here is to be able to infer the effects of a compound action from the atomic actions that compose it, and in particular, to be able to handle cases in which one of the composing actions is interfered with. It may be that the fluent `brokenOven` holds, which would prevent the `cookFood` action to occur, meaning that the `prepareLunch` action will not have its expected outcome. The event calculus can deal with this and ensure all the effects of the composing actions hold by employing the `clipped` predicate that denotes those fluents that have been terminated.

Concurrent actions Some actions, taken together, might have cumulative and cancelling effects. For example, lifting up a dish from one side will have the effect of spilling its contents.

However, lifting both sides together will have the effect of getting the dish off the table and keeping it full. The challenge here is to be able to obtain from the actions `liftFromRight` and `liftFromLeft` the fluents `offTable` and ¬`spilled`, even though the individual effects of those actions are ¬`offTable` and `spilled`. To address this, the event calculus can integrate a `cancels` predicate that describes the relations holding between two actions when they are done at the same time, and by enabling fluents to be determined by two simultaneous event occurrences rather than one [219].

The event calculus is described in further detail in section 6.1.1, as it is the formalism we chose to base our model of action on.

3.2 Comparing the Event-Calculus to Available Formalisms

3.2.1 The Situation Calculus

The situation calculus was first introduced by John McCarthy in 1963. It was designed for representing and reasoning about dynamical domains operating from determined situations [166, 200]. These situations correspond to a possible world history, or sequence of actions. These can be thought of either as a time point or as the set of all sentences, namely the theory, that hold at a given time point [140]. The reserved binary symbol `do(a,s)` represents the situation that results from performing action a in situation s. Consider an example in which Adam gives a signal to Eve for shooting Chris, ensuing in Chris's death. This can for example be modelled as *do(die(Chris),do(shoot(Eve,Chris),do(signal(Adam),s)))*.

There are two types of fluents in the situation calculus, which both take a situation as their final argument. *Relational* fluents are properties of the world that can be true or false, such as the fact that Chris is harmed, while *functional* fluents return a situation dependent value, such as the distance between two objects in a given situation [116, 140]. Properties of a situation s are described by the *uniform formula* in s. A formula is uniform in a given situation s if it does not mention any other situation besides s, so that it can be evaluated with regard to situation s only [215].

As such, the event calculus is similar to the situation calculus in that, starting from an initial situation, a finite sequence of actions is performed within the modelled world. These actions may have preconditions, meaning that some of these are not executable in a given situation. However, the event calculus differs from the situation calculus in that it is narrative-based, that it, it assumes a time structure that is independent of any occurrences [173]. Where the situation calculus initially aims to represent hypothetical actions and situations, the event calculus was designed primarily to reason about actual events and the resulting times at which fluents hold. Where the situation

calculus aims to allow reasoning over transitions between global situations, the event calculus aims to allow reasoning over intervals of time.[1] This means that while the situation calculus functions well when a single agent performs instantaneous and discrete actions, it does not attain the expressiveness of the event calculus when having to deal with actions that have duration and that might overlap with each other [208].

3.2.2 The Fluent Calculus

The fluent calculus is a variant of the situation calculus which finds it origin in the formalism put forward by Hölldobler and Schneeberger in 1990 [114]. Its semantics are best described in [234], keeping in mind that symbols for predicates and functions begin with capital letters, whereas variables are written in lower case:

> Central to the axiomatization technique of the Fluent Calculus is a function *State(s)* which relates a situation s to the state of the world in that situation. In turn, these world states are collections of fluents, which are reified to this end, i.e. treated as terms. That is to say, we use fluent *terms* like *On(A,Table)*, where *On* is a binary function symbol. Fluents that are known to hold in a state are joined together using the binary function symbol "$\circ$ ". This function is assumed to be both associative and commutative. It is illustratively written in ifix notation. Associativity allows us to omit parentheses in nested applications of $\circ$.
> As an example, suppose that about the initial situation S_0 in some Blocks World scenario it is known that block A is on some block x, which in turn stands on the table, and that nothing is on top of block A or block B. Using the Fluent Calculus, this incomplete knowledge can be axiomatized by a first-order formula as follows:
>
> $$\exists x, z[State(S_0) = On(A, x) \circ On(x, Table) \circ z$$
> $$\bigwedge \forall y, z\prime[z \neq On(y, A) \circ z\prime \bigwedge z \neq On(y, B) \circ z\prime]]$$
>
> Put in words, of state *State(S_0)* it is known that for some x both *On(A,x)* and *On(x,Table)* are true and possibly some other fluents z hold, too – with the restriction that z does not include a fluent *On(y,A)* nor a fluent *On(y,B)*, of which we know they are false for any y.

As such, the fluent calculus essentially differs from the situation calculus in that situations are here considered as representations of states rather than as histories of action occurrences. We therefore

[1] In the original event calculus these time intervals are explicitly represented, though in later simplified versions they are only implicitly available [240].

chose the event calculus over it for similar reasons, stemming from the greater expressiveness of the event calculus pertaining to event occurrences and time intervals. We now turn to a discussion of the frame problem, which all systems for reasoning about actions and change must address, and review how each of the three calculi appraised here undertake to answer it.

3.3 The Frame Problem

This problem, introduced by McCarthy and Hayes in 1969 [166], describes the challenge for a logical system to account for all the things that stay the same within a knowledge base, without having to explicitly describe a large number of intuitively obvious absences of change. It corresponds to the requirement of finding an efficient way of handling "non-change", i.e., of deriving the effects that actions do not have. The term to describe it is borrowed from the concept of "frame of reference" in physics that denotes the assumed stationary background with respect to which motion is measured [208]. It is best understood as two related problems: the *representational* frame problem, which denotes the requirement to specify all the non-effects of actions, and the *inferential* frame problem, which pertains to the process of actually inferring them. For example, we may need to both represent and derive that, when an agent moves an object, its own location does not change. It is important to find an operative solution to the frame problem because most things stay the same most of the time, since each action only changes a fraction of all fluents [208]. One approach to addressing the problem consists in providing explicit *frame axioms* whose role is to individually state what indeed stays the same. However, this is costly and sometimes impossible to implement, since there can theoretically be an infinity of things that can remain unchanged: if there are X number of fluent predicates and Y number of actions, then $O(XY)$ frame actions will have to be stated.

The event and situation calculi employ a similar method for addressing the frame problem. Relative to the situation calculus, Ray Reiter introduced a variant of the framework containing axioms which, rather than stating the effects of each action, consider instead how each fluent evolves over time [198]. These are called *successor state axioms*. They posit that, following a possible action, a fluent is true in a resulting state *if and only if* the action's effect made it true *or* it was true already and the action did not affect it. This means that these axioms specify the truth value of a fluent in a resulting state as a function of the action and of the fluent's truth value in the original state. The resulting state is therefore entirely specified by the previous state and does not require additional frame axioms [208]. Because this solution focuses on values rather than on the effects of actions, it dramatically reduces the number of axioms required. In the event calculus, the frame problem is addressed by resorting to axioms which constrain the value of fluents in the same way as successor state axioms do. The law of inertia is imposed by rules stating that fluents are true if they were

already true at a previous time point and not terminated by the occurrence of an action with that effect. In addition, in the event calculus, predicate completion is necessary in order to obtain that an action is performed only if it is explicitly stated that this is so, and that a fluent is true only if an action which makes it true has been performed (or if it is initially true).

The fluent calculus, created in part to resolve it, tackles the frame problem by taking advantage of the fact that it employs first-order logic terms instead of predicates to represent states, a process know as *reification* [234]. Where a predicate represents a condition that can be true or false when evaluated over a set of terms, a term in first order logic represents a (more or less complex) *object*. The problem is then solved by specifying the effects of actions though statements about how a term representing a state changes after an action is executed: the state after the execution of an action is identical to the one before but for the conditions changed by the action. This induces that, at any given state, a unique successor state axiom is sufficient to infer the totality of the changes brought about by a given action (as demonstrated in [233]).

3.4 Converting Logical Languages into Logic Programs

Logical languages for reasoning about actions and change can be converted into logic programs via algorithms that act as translational operators. This allows well established techniques for proving properties of logic programs to be applied to these languages and presents a number of advantages [140]. In particular, this type of conversion facilitates implementation, thereby enabling straightforward testing, tweaking, or modulation. It also grants access to negation as failure, whose use provides a variety of robust and useful semantics for default reasoning. It is possible to convert languages for reasoning about actions and change into logic programs both by appealing and by not appealing directly to causal theories; some do (for example Prop 6.1 from [165]), while some do not (for example [153]). In the following section, we introduce the features and type of logic programming we chose to implement our language in.

Chapter 4

State of the Art: Logic Programming

4.1 An Introduction to Logic Programming

Logic programming is a programming technology based on formal logic. It is a paradigm that hinges around the notion that systems can be constructed by expressing knowledge in a particular formal language, and that problems can be solved by running inference procedures on that knowledge. Logic programs consist in a set of sentences written in logical form, that express the facts and rules of a problem domain. Rules are written as clauses of the form: $A \leftarrow A_1, \ldots, A_n$ where each $A_i \in Atom$ is an atomic formulae. They are read declaratively as logical implications: "A if A_1 and... and A_n." The atom on the left side of the implication is called the head of the rule, and the atoms on the right side of the implication are called the body. Facts are rules that have no body, and are written in the simplified form: A. When modelling artificial logical agents, as is the case here, these facts and rules represent the knowledge base of the agent from which the agent reasons and plans. In the simplest case of classical logic in which $A, A_1, \ldots, A_n$ are all atomic formulae (i.e. positive predicates), these clauses are called definite or Horn rules. As we will see, there are other types of clauses, in particular some that allow for the conditions in the body of the clause to be negations of atomic formulae.

4.2 Non-Monotonic Reasoning and Negation as Failure

Non-monotonic logic has been put forward by artificial intelligence researchers as a way to handle the kind of defeasible generalisations that pervade much of our commonsense reasoning and that are poorly captured by classical logic systems [117]. The term covers a family of formal frameworks devised to apprehend the kind of inference where no conclusion is drawn definitely, but stays open to modification in the light of further information. On a regular basis, it seems, we draw conclusions from bodies of data that can be easily dropped when faced with new data. For example, if we are told that *Tweety is a bird*, a natural inference we would make is that *Tweety can fly*. However, if we were to learn that *Tweety is a penguin*, we would drop this conclusion. Importantly, this kind of default based reasoning is prominent in ethical reasoning. Indeed, we may determine the moral value of an action, for example theft, differently depending on surrounding information. Such factors as the presence of alternative options, indirect consequences, or extenuating circumstances might overthrow our ethical judgement: while it is commonly agreed upon that theft is wrong, there are cases in which we might want to say that thieving was the right action. Accordingly, non-monotonic goal specification languages are particularly well suited to modelling ethical reasoning. Negation as failure is a form of negation added to Horn rules that allows for non-monotonic reasoning. This inference rule introduces the negative literal *not P*, which can be "proved" true just in case the proof of P fails [208]. In other words, it allows us to assume that something is false as long as we cannot prove that it is true, in a form of default reasoning akin to the closed-world assumption. In practise, we use the operator "*not*" to distinguish negation as failure from strong negation denoted by "¬".

We saw in section 3.3 that the frame problem challenges us with the necessity to succinctly capture the non-effects of actions. This amounts to formalising the default assumption (known as *the common law of inertia*) that a fluent stays constant *unless* there is evidence to the contrary. In classical logic, the main handicap for dealing with this stems from its inherent monotonicity: the set of conclusions that can be drawn from a set of rules necessarily increases with the addition of new rules, meaning that it cannot handle rules – like the commonsense law of inertial – that contain an open-ended set of exceptions. Non-monotonic logic, reversely, fundamentally allows such exceptions. More precisely, negation as failure allows for circumscription within the event calculus and allows us to handle the frame problem via rules of the following type:

```
clipped(S,P,T):-terminates(S,E,P,T).
holds(S,P,T+1):-holds(S,P,T),not clipped(S,P,T),[...].
```

With S a simulation, P a positive fluent, T a time point and E an event. These axioms play together the role of successor state axioms and provide a robust solution to the problem in that it holds in

the presence of complex situations, such as the occurrence of concurrent actions or actions with indirect effects.

4.3 Answer Set Programming

Answer Set Programming is a form of declarative logic programming with negation as failure which is suited for representing Artificial Intelligence problems, particularly those that relate to knowledge representation and automated reasoning with incomplete information. Also called Disjunctive Logic Programming under the stable model semantics (DLP), it has blossomed from the works of Gelfond & Lifschitz [81, 83] as well as Minker [162]. With roots in both Prolog and fast propositional satisfiability provers, it unifies previous non-monotonic reasoning formalisms [152, 208]. It works by converting logic programs – encoded as extended disjunctive programs – into ground form, then using propositional model checking techniques to extract stable models. These stable models, or answer sets, declaratively identify the solutions to given problems [79]. They are based on both available and unavailable information, and form a coherent set of assumptions that describe one rational viewpoint inferred from given rules and facts. Each stable model is a minimal set of atoms representative of information justified by rules in which the head of the rule is the atom and where each literal in the body is satisfied. We give here a succinct overview of the answer set semantics for a program defined over a set of literals *Lit* (see [81] for more details).

First, we define the syntax of extended disjunctive program. Such programs use both explicit negation $\neg$ ($\neg p$ meaning that it can be proven that p is false) and negation as failure *not* (*not* p meaning that p cannot be proven to be true). An *atom* is an expression *p(t1, . . .,tn)*, where p is a predicate of arity n and *t1,. . . ,tn* are terms. A *literal* L is either an atom p or its explicit negation not $\neg p$. Two literals are said to be complementary if they are of the form p and $\neg p$ for some atom p. A set S of literals is said to be consistent if, for every literal L $\in$ S, its complementary literal is not contained in S [32].

Definition 1 (Extended disjunctive program)**.** An *extended disjunctive program* (EDP) is a set of *rules* of the form:

$$L_1; \cdots ; L_l \leftarrow L_{l+1}, \ldots, L_m, not\, L_{m+1}, \ldots, not\, L_n \ (n \geq m \geq l \geq 0)$$

where each $L_i \in Lit$ is a literal, namely, A or $\neg A$ for an atom A.

Negation as failure (NAF) is denoted by *not*, and $not\, L$ is called a *NAF-literal*. The symbol ";" represents disjunction. For each rule r of the above form, $head(r)$, $body^+(r)$, $body^-(r)$, and

$notbody^-(r)$ denote the sets of NAF-literals $\{L_1, \ldots, L_l\}$, $\{L_{l+1}, \ldots, L_m\}$, $\{L_{m+1}, \ldots, L_n\}$, and $\{not\, L_{m+1}, \ldots, not\, L_n\}$, respectively.

Definition 2 (Integrity constraint, Fact, Normal and positive rules)**.** A rule r is an *integrity constraint* if $head(r) = \emptyset$; r is a *fact* if $body(r) = \emptyset$; r is a *normal rule* if $l = 1$ (that is $head(r) = \{L_1\}$); and r is a *positive rule* if $body^-(r) = \emptyset$.

A program P with variables is semantically identified with its ground instantiation. The semantics of EDPs is given by the *answer set semantics* [81]. Simple models are defined through satisfaction as follows, denoting by Lit the set of all grounded literals.

Definition 3 (Satisfaction)**.** A set $S \subseteq Lit$ *satisfies* a rule r iff $body^+(r) \subseteq S$ and $body^-(r) \cap S = \emptyset$ imply $head(r) \cap S \neq \emptyset$. S is a *model* of a ground program P iff S satisfies every rule in P.

However, answer set, also called stable models when explicit negation is not used in the program, requires additiona conditions, which amount to ensure that each literal present in an answer set produced by a rule. We first define answer set for the restricted case of a *positive program* : a program P containing only positive rules ($\forall r \in P, body^-(r) = \emptyset$).

Definition 4 (Answer set of a positive program)**.** A set $S \subset Lit$ is a consistent *answer set* of P iff S is a set such that

(i) S is a model of P, that is, it satisfies every rule from the ground instantiation of P, and
(ii) S is consistent, that is, it does not contain a pair of complementary literals L and $\neg L$.
(iii) S is minimal for set inclusion, that is it does not contain any subset S

Note that if P is a positive normal program (a program containing only positive normal rules), then there is a unique stable model, verifying condition (i) and (iii), which can be obtained by forward chaining (though it might be inconsistent).
Next, let P be any EDP and $S \subseteq Lit$. We first define the *reduct* of P wrt S, which allows to project P to a positive form.

Definition 5 (Reduct)**.** The *reduct* P^S of a program P with respect to a set of literals S is defined as the set of all rules r^S : $head(r) \leftarrow body^+(r)$ where r is a rule in the ground instantiation of P such that $body^-(r) \cap S = \emptyset$.

The reduct of P wrt S can thus be build by first removing all rules r of P whose negative body contains a literal in S and then removing the negative body of all the remaining rules. The answer of an EDP is then defined as follows.

Definition 6 (Answer set of an EDP)**.** S is an *answer set* of P iff S is an answer set of the positive program P^S.

Answer Set Programming offers well-defined semantics and efficient operational solvers, allowing for effective automated demonstrations and problem solving, but its foremost advantages over other planners stem of its degrees of flexibility and expressiveness. Indeed, because the planning operators and constraints can be expressed in terms of fully declarative logic programs, they are not restricted to the finite format of any particular planning formalism [208]. It has extensive expressive power because it allows the modeller to formally express every property of finite systems over a function-free first-order structure; in which the decision-making process is solvable in non-deterministic polynomial time. This means it permits the encoding of programs which are not fit to be translated to SAT in polynomial time [32]. The established and relatively straight-forward translation of the event calculus into Answer Set Programming as done for instance in [135] and [144] also motivated our choice of appealing to it. However, its high expressiveness means that ASP functions at a high computational costs, meaning that the implementation of complex and efficient systems can be problematic. Some work has been done in order to address this issue and present more efficient frameworks (e.g. [124, 154, 221]), leading to real-world applications (e.g. [58, 123, 204, 205]). Other works have focused on theoretical characterisations of ASP, language extensions and the optimisation and evaluation of algorithms (see [32] for a review).

Chapter 5

Contribution: Structural Scheme

5.1 Review of Process

So as to give the reader an idea of the process and questionings that eventually led to the present work, we here summarise the different stages of enquiry.

Stage 1 In order to begin tackling a topic as wide as "*models of ethical reasoning*", we chose to first focus on modelling a specific dilemma and accompanying theory. We chose the *trolley problem* dilemma and the doctrine of double effect, for their ubiquity in the philosophical literature and for the challenges that they present relative to notions of causality. The main difficulty faced at this stage stemmed from modelling the notion of *prevention*, which involved having to reason over events that in fact never occurred in the simulation. This difficulty also arose from the choice of appealing to the event calculus, within which scenarios are represented as a succession of events and world states, rather than as transitions between outcomes that do not explicitly distinguish event occurrence from event non-occurrence. We implemented this using Answer Set Programming, as is the case throughout. These formal choices were made in order to produce a framework that could most closely represent the diversity of mechanisms at play in judgements of causal and moral responsibility, that could explicitly distinguish situational from ethical aspects so as to not confound reasoning processes that are different in nature, and that could permit the comparison of various ethical theories. This initial work resulted in a first publication [30].

At this point, the features of the framework are the following:

- 1 answer set = 1 scenario,
- multiple scenarios can be modelled but not compared,

- any number of actions can be possible at t=0,
- one action is performed per scenario at t=0,
- the agent can test the compatibility of its action against one ethical theory,
- there is an explicit representation of causing and preventing.

Stage 2 The next stage consisted in extending the framework to model more ethical theories. Two main types of relevant theories were distinguished, consequentialist and deontological. The different computational demands made by these two types were identified and addressed. One challenge consisted in allowing the agent to compare multiple possible actions available at the same time so as to chose the best one, as is required by some consequentialist principles (rather than just testing the compatibility of each action against a theory, as done previously). This led us to introduce the notion of *simulation* noted s_n, which allowed the agent to test out the effects of different actions upon an identical initial situation, within a single launch of the program, and compare their effects inside a single answer set. For example, in the trolley problem, the agent performed the *push* action in s_1 and the *switch* action in s_2, and a comparison of these two simulations showed that they were equivalent in consequentialist terms (i.e. one dead, five saved). The choice of which action was performed in which simulation was stated in the program in the form of facts. Another issue raised by consequentialist comparisons involved giving meaningful weights to events, or in other words, producing workable theories of the Good. Finally, the recognition of distinct processes led to the division of the framework into four parts: an *action model*, a *causal model*, a *model of the Good* and a *model of the Right*, resulting in better hierarchical organisation and openness to modularity. This work resulted in a second publication [28].

At this point, the features of the framework are the following:

- 1 answer set = multiple scenarios,
- multiple scenarios can be compared,
- any number of actions can be possible at t=0,
- one action is performed per scenario at t=0,
- the agent can test the compatibility of its action against multiple ethical theories.

Stage 3. The third stage saw a number of additions. First, we updated the framework to allow for reasoning over plans of actions rather than over single actions. This led us to model inaction, as the choice of refraining to act was considered morally meaningful within these plans. It also prompted us to describe and model new causal relationships, such as making possible an agent's actions (enabling), which was distinguished from making possible an automatic event (causing). Second, we explored properties related to causality and agent responsibility in these plans. The

aim was to determine some of the characteristics that a relationship between an *affector* and an *end-state* can display. For example, assuming that affector ϕ causes outcome ψ, if ϕ hadn't occurred, was there any other way for the agent to make ψ true? In other words, was ϕ crucial to ψ? In order to answer such a question, the agent needed to have a way to generate all possible versions of a particular domain, i.e. all combinations of possible actions and omissions given an initial situation. To do this, we appealed to simulations again, each one now corresponding to a possible set of performed actions (this process could also be guided to generate not all, but a subset of possible scenarios). In stage 2, the choice of which action to do in which simulation was stated, in this third stage, plans were generated automatically. But this generation proved difficult and computationally costly, in particular because all was done within a single launch of the program and resulted in a single answer set. Also, for reasons linked to combinatorial explosion, the framework could only process domains in which only zero or one action was ever *possible* at each time point. If not, the generated simulations could be faulty. This process nevertheless allowed us to model four different causal properties, under the heading of what we called *scenario-based* causality - as opposed to *event-based* causal properties like causing and enabling. These updates resulted in a third publication [29].

At this point, the features of the framework are the following:

- 1 answer set = multiple scenarios,
- multiple scenarios can be compared,
- up to one action can be possible per time point,
- up to one action can be performed per time point,
- any number of actions can be performed in a scenario,
- there is an explicit representation of omissions,
- there is an explicit representation of *event-based* causality,
- there is an explicit representation of *scenario-based* causality.

Stage 4 The fourth stage essentially consisted in fully combining stages 2 and 3, i.e. allowing the agent to apply ethical theories to plans of actions. This process was far from trivial, and brought to light many new issues, such as the challenge of assigning moral weight to single actions within wider plans, or assigning weight to plans as a whole. It also made salient the fact that in the philosophical literature, ethical theories are typically formulated to tackle individual actions rather than plans, even though actions in the real world are arguably always part of plans. This led us to amend or extend some of these theories as we modelled them.
In order to better handle problems related to combinatorial explosion and processing time, we also divided the computational process into two steps. Instead of generating different simulations inside

Table 5.1: Summary of process

Features	stage 1 [30]	stage 2 [28]	stage 3 [29]	stage 4
number of scenarios per answer set	1	∞	∞	1, then ∞
number of actions possible per time point	∞	∞	up to 1	∞
number of actions performed per time point	1 at t=0	1 at t=0	up to 1	up to 1
number of actions performed per scenario	1	1	∞	∞
number of modelled ethical theories	1	8+	8+	8+
comparison of available actions	•	✓	✓	✓
comparison of available plans	•	•	✓	✓
representation of omissions	•	•	✓	✓
representation of causing/preventing	✓	✓	✓	✓
representation of enabling/excluding	•	•	✓	✓
representation of scenario-based causality	•	•	✓	✓

∞ for any number, ✓for yes, • for no

one answer set, we began by generating one answer set per version of a scenario, using the first of the four models described above, the *action model*. Through an external script, we launched this process then recuperated the resulting answer sets, cleaned them and edited them so that each different answer set was made to correspond to a numbered simulation. The script then launched the edited answer sets together with the remaining three models to yield final results. This two-step procedure effectively amounted to a less costly method than the one used in stage 3 for allowing the agent to get a representation of all possible versions of a scenario. It also enabled the framework to handle scenarios in which multiple actions were possible at the same time - though the agent still only *performed* up to one action per time point, but this was a simplifying design choice rather than an inherent limitation of the framework. At this point, the features of the framework were, and still are, the following:

- 1st step of the launch generates multiple answer sets where 1 answer set = 1 scenario,
- 2nd step of the launch generates 1 answer set where 1 answer set = multiple scenarios,
- multiple scenarios can be compared,
- any number of actions can be possible per time point,
- up to one action can be performed per time point,
- any number of actions can be performed per scenario,
- there is an explicit representation of omissions,
- there is an explicit representation of *event-based* causality,
- there is an explicit representation of *scenario-based* causality,
- the agent can test the compatibility of volitions and plans against multiple ethical theories.

We summarise these four stages in table 5.1.

5.2 Review of Framework Attributes

We review here the characteristics that the resulting framework has and doesn't have.
As such, the model:

- is top-down,
- is generic, such that any number of theories and concepts can be modelled within it (ethical aspects are domain independent),
- is implemented,
- is explicit,
- is rationalist, in the sense that it does not purport to model emotions,
- is ethics agnostic, such that it makes no recommendations on what ethical theories to apply where,
- is not embedded, such that it is not implemented in a physical entity like a robot,
- does not support intentional or epistemic states.

The world:

- is fully observable,
- is closed, so that if a fluent is not true then it is false,
- evolves through serial action execution orders,
- is represented by world states and events rather than transitions,
- is partly deterministic: the effects of all events are deterministic but, given a moment in time, the volition choices that agents make in the future can be non-deterministic (as discussed in section 9.1, they are non-deterministic relative to principles modelled *in foresight* and deterministic relative to principles modelled *in hindsight*),
- uses discrete time,
- is time agnostic, such that no difference is made between events close in time and events far in time.

The agent:

- is omniscient, with a perfect internal model of infallible actions and world dynamics,
- is rational,
- can interact with other pseudo-agents inside scenarios.

We now turn to a presentation of the four models that compose this framework.

5.3 Models and Modularity

The aim of producing an explicit representation of ethical reasoning is to enable an agent to assess the permissibility of an action or set of actions, either to inform its own decision-making or to judge the behaviour of others. To achieve this, we make it possible for the agent to "test out" possible actions in specified simulations so as to evaluate their consequences or inherent ethical status. The outcome of the simulation then yields a set of permissible or impermissible actions, which dictates its upcoming behaviour or appraisal of other agents. The framework presented here is concerned with this assessment process, rather than with what the agent chooses to do with its upshot.

The ethical decision-making process is apprehended as a four-step procedure captured by four types of interdependent models: an *action model*, a *causal model*, a *model of the Good*, and a *model of the Right*. The first two models provide the agent with an entirely ethics-free understanding of the world, the second two provide an ethical over-layer that the agent can parse and apply back onto its knowledge of the world. We define these models here, as illustrated in figure 5.1.

Definition 7 (Action model)**.** An *action model* $\mathbb{A}$ enables the agent to represent its environment and the changes that take place in it. It takes as input a *set of scenarios* $\mathbb{S}$ which defines the set of considered actions and omissions. It is composed of an *initial situation* containing the fluents that hold at t=0, a *specification of events* containing a set of events and of dependence relations, and an *event motor* which enables the simulation to evolve. It generates an *event trace* of each simulation, which designates for each time point the events that occur and fluents that hold.

Definition 8 (Causal model)**.** A *causal model* $\mathbb{C}$ tracks the causal powers of actions, enabling reasoning over agent responsibility and accountability. It takes as input the *event trace* given by the *action model* and a *specification of events* containing a set of events and of dependence relations. It is composed of a *causal motor* which enables the creation of the causal tree that characterises the simulation. It generates a *causal trace* of each simulation which designates for each time point the causal relations that exist between events and fluents.

Definition 9 (Model of the Good)**.** A *model of the Good* $\mathbb{G}$ makes a claim about the intrinsic value of goals or events. It is composed of a *specification of targets*, a *specification of modalities*, an *ethical specification of events* which describes the extent to which events affect different modalities, and a *theory or set of theories of the Good*. It generates a *goodness assessment* of events, made explicit by a valuation of events as having good or bad ramifications.

Definition 10 (Model of the Right)**.** A *model of the Right* $\mathbb{R}$ considers what an agent should do, or is most justified in doing, within the circumstances of its actions. It takes as input the *causal trace* given by the *causal model* and, in the case that a given theory of the Right contains consequentialist principles, a *goodness assessment* given by the *model of the Good*. It is composed of a *theory or set of theories of the Right*, and, in the case that a given theory of the Right contains deontological principles, a *set of deontological specifications*. It generates a *rightness assessment* of actions, made explicit by a valuation of actions as permissible or impermissible in relation to each given theory of the Right.

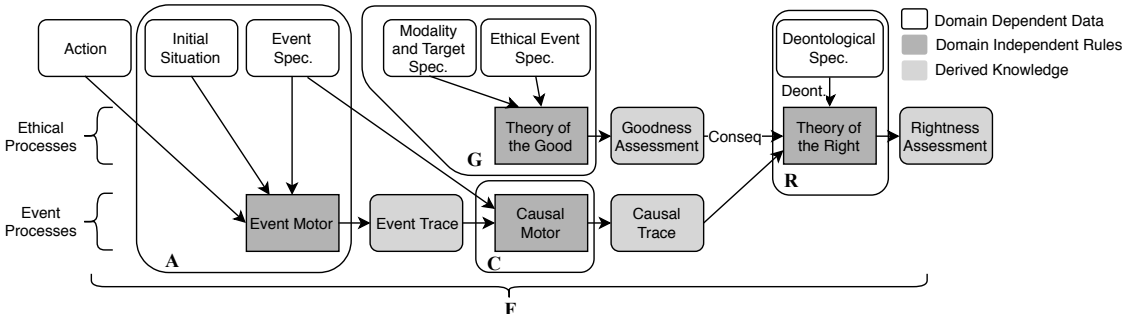

Figure 5.1: Models and modularity

This modular scheme coincidentally roughly aligns with philosopher James Gips's recommendations for modelling an artificial agent capable of reasoning consequentially [86], which are that the agent must have:

(1) A way of describing the situation in the world,
(2) A way of generating possible actions,
(3) A means of predicting the situation that would result if an action were taken given the current situation,
(4) A method of evaluating a situation in terms of its goodness or desirability.

The four types of models are interdependent at varying degrees. Models of the Good and the Right always rely on an action and a causal model. But while a causal model is always necessary, the particular formulation of the causal motor may vary, to account for instance for different definitions of causes and consequences. Because the event motor provides the basis for the framework, however, it is proposed as unique and unvarying. Pertaining to ethical models, having a model of the Good is necessary to model consequentialist theories of the Right as well as deontological ones with consequentialist constraints, but not purely deontological ones. Inter-dependencies may also hold within a type of model, particularly in the case of theories of the Right which call upon one-another.

The well defined hierarchy between the different types of models gives the framework the capacity to model but also to compare a potentially unlimited number of ethical theories. Compartmentalising different types of processes means they can be analysed specifically. Substituting a particular model while keeping constant the others allows for the individualised examination of its ramification. Based on these models, we may now define the framework which enables the ethical assessment of actions.

Definition 11 (Ethical assessment framework)**.** The *ethical assessment framework* is defined as:

$$\mathbb{F} = \langle \mathbb{A}_i, \mathbb{C}_i, \mathbb{G}_i, \mathbb{R}_i \rangle$$

Given an ethical assessment framework $\mathbb{F} = \mathbb{F} = \langle \mathbb{A}_i, \mathbb{C}_i, \mathbb{G}_i, \mathbb{R}_i \rangle$, a set of scenarios $\mathbb{S}$ (as defined in section 6.2), a set of considered volitions $\mathcal{I}_c$ and a given ethical principle h, we then define the set of permissible volitions as

$$Permissible(\mathbb{F}, h, \mathcal{I}_c) = Th_{\mathtt{per}(j,\mathcal{I}_c)}(\mathbb{S} \cup \mathbb{A}_i \cup \mathbb{C}_i \cup \mathbb{G}_i \cup \mathbb{R}_i)$$

Part III

Action Model

Chapter 6

Contribution: An Action Model

6.1 The Power to Act

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 or let our pet die of hunger. Responsibility therefore also pertains to the power to *not* act.

Whether there is a fundamental moral difference between actions and omissions is an important point of debate within moral philosophy (e.g. [24, 73, 128]) as well as within behavioural psychology (e.g. [18, 19, 60]). 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. Philippa Foot's version of the doctrine of doing and allowing is one such answer. She argues that doing harm is worse than allowing harm by appealing to a distinction between *positive rights* (rights to be aided) and *negative rights* (rights not to be harmed). These rights respectively result in duties to aid and duties of non-interference, such that violating someone's negative right not to be harmed is to *do harm*, while violating someone's positive right to be aided is to *allow harm*. Foot then argues that negative duties are more important than positive duties. This results in the claim, for example, that it is worse to kill a child (infringement of the negative right not be harmed) than to let him drown (infringement of the positive right to be aided). James Rachels, reversely, argues that there is no fundamental moral difference between doing and allowing. Consider the following cases:

> (*Smith*) would gain a large inheritance if anything were to happen to his young cousin. One evening while the child is taking a bath, Smith sneaks into the bathroom and

> drowns him.
> (*Jones*) also would gain a large inheritance if anything were to happen to his young cousin. One evening while the child is taking a bath, Jones sneaks into the bathroom to drown him. However, just as he enters, his cousin slips, hits his head, falls face down into the water. Jones stands by, ready to push his cousin's head back under if necessary. It is not necessary and the child drowns.

The argument then goes that if indeed there was a fundamental moral difference between doing and allowing, we would think that Smith is worse than Jones, yet, arguably, this is not the case [194]. A reply to this objection is that it is only necessary to find a single case in which doing harm is unambiguously worse than allowing the exact same harm to be able to say that there is a moral difference between the two – even if this is not true in *all* cases. Real world situations might here provide some compelling examples. Consider for instance this decision made by Winston Churchill during the second World War discussed by Michael S. Moore in [177]:

> The British had broken the German coding device called Ultra, and learned that the Germans planned to bomb Coventry to dispirit the British populace. Churchill could have prevented the killings of the citizens of Coventry by alerting them; but this also would have tipped off the Germans that the British had obtained the German coding-device. Churchill justified his not saving the citizens of Coventry by his thereby saving many more British lives in a shorter, more successful war effort. His socialist War Cabinet member urged him not to "play God" with people's lives. Yet given the consequences, wasn't Churchill right? Whereas if Churchill had had to kill the citizens of Coventry in order to maintain the British advantage in intelligence – say some Coventry citizen was a spy who learned this but it is unknown which citizen it is – surely we may not kill to achieve the same good consequence.

Whichever perspective we embrace over whether omissions are less morally significant that actions, this debate is a testament to the fact that it is critical to be able to model them 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. We resolve that every time an action or more is available but no action is performed, an omissions occurs. Throughout, we refer to actions and omissions as *volitions*.

6.1.1 Basic Action Model

The basic *action model* is the fundamental building block of our framework, it indicates the way in which the dynamics of our system will be expressed. It corresponds to the full event calculus described in [219], with a number of additions.

Discrete time For the purpose of simplicity and to later make the study of causality clearer, discrete time is employed, and is represented by integers.

Fluent inertia We distinguish inertial fluents from non-inertial fluents [173]. Once initiated by an event occurrence (or if initially true), inertial fluents remain true until they are terminated by another event occurrence. Non-inertial fluents are only true at the point in time at which they have been initiated by an event occurrence, or at time t=0 if they were true initially.

Automatic events To fit the requirements of modelling ethical dilemmas pertaining to complex and realistic scenarios, one of our contributions has been to introduce automatic events in addition to actions. These automatic events occur when all their preconditions, in the form of fluents, hold, without direct input from the agent. Actions additionally require that the 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.

Simulations We also index each time-dependent predicate to a simulation, which associates theses predicates to a scenario. This indexing serves to investigate the hypothetical alternatives to a particular scenario, enabling the modelling of properties of causality as well as ethical theories. Simulations are noted s_n.

The dynamic domain is as follows:

$\mathcal{S}$ is a set of simulations (variables `S1,S2`...),
$\mathcal{G}$ a set of agents (variables `G1, G2`...),
$\mathcal{T}$ a set of time points (variables `T1,T2`...),
$\mathcal{P}$ a set of positive fluents (variables `P1,P2`...),
$\mathcal{F}$ a set of fluents containing all positive fluents and their negation (variables `F1,F2`...),
$\mathcal{U}$ a set of automatic events (variables `U1,U2`...),
$\mathcal{X}$ a set of action names (variables `X1,X2`...),
$\mathcal{A}$ a set of actions (variables `A1,A2`...),
$\mathcal{O}$ a set of omissions (variables `O1,O2`...),
$\mathcal{I}$ a set of volitions where $\mathcal{I} \equiv \mathcal{A} \cup \mathcal{O}$ (variables `I1,I2`...),

$\mathcal{E}$ a set of events where $\mathcal{E} \equiv \mathcal{I} \cup \mathcal{U}$ (variables `E1,E2`...).

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

We now describe how events are defined. Note in particular that an action subsumes an action name and the agent that performs the action, and depends on the capacity that an agent has to perform it. Not all agents in a domain will be able to perform all actions. In the *event specifications*, we can define other restrictions of this kind. For example, the preconditions for an action might differ across agents.

```
action(act(G,X)):-capable(G,X).
volition(A):-action(A).
volition(O):-omission(O).
event(I):-volition(I).
event(U):-auto(U).
```

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, effects and priority rules.

`prec(F,E)` indicates that F is a precondition to E.

`effect(E,F)` indicates that E can cause F. All events except omissions are to be *well defined* in the sense that if F is an effect of E, then neg(F) is a precondition to E, and if neg(F) is an effect of E, then F is a precondition to E. This is to avoid these events from making true what is already true.

`priority(E1,E2)` defines priorities between events to avoid incompatible events from occurring at the same time. It draws the distinction between the state of an event as possible and the state of an event as effectively occurring in a simulation. Note that, though we do not make use of this, it is possible to allow for non-determinism in the triggering order between two incompatible events E1 and E2 by indicating that they have symmetrical priorities (that is, E1 has priority over E2 and E2 over E1). This will result in the generation of different answer sets for each possible order of priority. If two events are compatible, on the other hand, they can occur simultaneously.

The *initial situation* is composed of fluents that are true initially.

`initially(F)` indicates that F is true at t=0;

`nonInertial(F)` points out the special kinds of fluents that are not constrained by the commonsense law of inertia.

Event Motor

An *event motor* is a set of domain independent axioms governing the dynamics of a scenario. It is composed of *event effect axioms* and *event precondition axioms*. *Event effect axioms* characterise the behaviour of fluents relative to the occurrence of events.

`initiates(S,E,P,T)` indicates that E occurs and initiates P at T in S (P is not the negation of a fluent).

`terminates(S,E,P,T)` indicates that E occurs and terminates P at T in S.

`clipped(S,P,T)` indicates that P is clipped at T in S, because it was just terminated by an event occurrence.

`holds(S,F,T)` indicates that F holds at T in S, with F either corresponding to a positive fluent P or its negation neg(P). This predicate traces the evolution of fluents across time inside a simulation.

These predicates enable us to axiomatize the principles that govern fluents: a fluent holds at T is S if it was initiated by an event occurrence at T-1 in S; a fluent which is true at T in S continues to hold until the occurrence of an event which 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. This last statement means that in the *initial situation* only positive fluents need to be stated, since if nothing is stated about a fluent, its negation will automatically hold.

```
initiates(S,E,P,T):-occurs(S,E,T),effect(E,P),positiveFluent(P).
terminates(S,E,P,T):-occurs(S,E,T),effect(E,neg(P)).
clipped(S,P,T):-terminates(S,E,P,T).
holds(S,P,0):-initially(P),sim(S).
holds(S,P,T+1):-initiates(S,E,P,T).
holds(S,P,T+1):-
    holds(S,P,T),not clipped(S,P,T),not nonInertial(P),positiveFluent(P),time(T).
holds(S,neg(P),T):-not holds(S,P,T),sim(S),positiveFluent(P),time(T).
```

Event precondition axioms characterise the behaviour of events relative to the truth values of fluents.

`complete(S,E,T)` indicates that E is complete at T in S, meaning that all the preconditions to E hold at T in S. Preconditions can either be positive fluents or their negation.

`possible(S,E,T)` indicates that E is possible at T in S, which means, if E in an automatic event, that E is complete, and if E is an action, that E is complete and has been performed by an agent.

`performs(S,A,T)` indicates that action A is performed at T in S. This predicate represents the agent's choice to perform an action, and only complete actions can be ever be performed.

`overtaken(S,E,T)` indicates that E is overtaken at T in S. This occurs at a given time when E is possible and there exists another event which is also possible and which has priority over it.

`priority(E1,E2)` indicates that E1 has priority over E2. Priorities are typically stated in the form of facts by the modeller.

`occurs(S,E,T)` indicates that E occurs at T in S. This happens when E is possible and not overtaken. Exactly one event (volition or automatic) occurs at each time point.

```
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),auto(E).
possible(S,A,T):-complete(S,A,T),performs(S,A,T),action(A).
:-performs(S,A,T),not complete(S,A,T).
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).
```

6.1.2 Action Model with Omissions

The meaningful fact of omitting to act only occurs when *acting is an option*. One cannot omit to act if there is no act to omit [246], as echoed in Aristotle's statement in the Nicomachean Ethics that "where it is in our power to do something, it is also in our power not to do it, and when the 'no' is in our power, the 'yes' is also" (1113b7-8) [10]. For this reason, we state that an omission occurs when at least one action is available (i.e. its fluents preconditions are all true) and no action is performed. But different omissions may occur multiple times inside and across simulations, and as such must be distinguished. Because an omission might occur when multiple actions are available, it is not viable to distinguish it from other omissions by indexing it to an action (e.g. `omit(G,X)` indexed to a not performed `act(G,X)`). We instead index each omission to the time at which it occurs and to the agent that carries it out, so that we may distinguish between the multiple omissions that might occur within one simulation. As such, an omission is denoted by `omit(G,T)` $\in \mathcal{O}$, and the simulation it occurs in is traced by the S in `occurs(S,omit(G,T),T)`. To prevent the same omission from occurring repeatedly, an action is not considered complete just after having been omitted, even while its preconditions are still true. To take omissions into account, the *event motor* is updated by adding the following rules and removing the previous definition of `complete`.

`complete(S,U,T)` indicates that automatic event U is complete at T in S, that is, all the preconditions to U hold at T in S.

`omitted(S,A,T)` indicates that action A is omitted at T in S. This is true when A is complete but not performed at T in S and that an omission occurs instead (meaning that any other action that might be complete at T in S is also not performed).

`complete(S,A,T)` indicates that action A is complete at T in S. This is true when all the preconditions to A hold at T in S, and A was not already just omitted.

`complete(S,O,T)` indicates that omission O is complete at T in S. This is true if at least one action is complete at T in S.

`possible(S,O,T)` indicates that omission O is possible at T in S. This is true at T in S when O is complete and no action occurs.

`priority(O,U)`) indicates that omission O has priority over automatic event U. This predicate ensures that an omission takes priority over any automatic event that is complete at the same time, allowing omissions to occur in the simulation at the relevant time and to stop events from occurring simultaneously, which may have confounding effects.

```
complete(S,U,T):-{not holds(S,F,T):prec(F,U)}0,sim(S),auto(U),time(T).
actName(X):-capable(G,X).
complete(S,act(G,X),T):-
    {not holds(S,F,T):prec(F,act(G,X))}0,not omitted(S,act(G,X),T-1),sim(S),time(T),
    action(act(G,X)).
omitted(S,act(G,X),T):-complete(S,act(G,X),T),occurs(S,omit(G,T),T).
complete(S,omit(G,T),T):-complete(S,act(G,X),T).
possible(S,omit(G,T),T):-
    complete(S,omit(G,T),T),{occurs(S,act(G,X),T):action(act(G,X))}0.
priority(omit(G,T),U):-occurs(S,omit(G,T),T),complete(S,U,T),auto(U).
```

In terms of dynamics, an omission makes true what the actions that were complete at the same time would have made false, and false what they would have made true. This amounts to considering that omissions cause what already exists but could have been made not to exist. In other words, if an agent chooses not to act, it is responsible for the parts of the world it could have changed. This fits with the intuition that people cannot be held responsible for that over which they have no power. This feature presupposes *well defined* actions that do not make true what is already true, and false what is already false. For example, the action to `stop` one's car is not possible if the car is already stopped. Considering that this action has the effect of negating the fluent `motion`, if we allowed for it to occur when `neg(motion)` was already true, then an omission would have the effect of initiating `motion`, i.e. starting the car again, which is inadequate. Moreover, because an omission

occurrence takes up one moment in time and because non-inertial fluents hold for one moment in time, these fluents must be carried onto the following time point to ensure that the state of the world does not change before and after an omission occurrence.

```
holds(S,P,T+1):-occurs(S,omit(G,T),T),holds(S,P,T),not clipped(S,P,T),nonInertial(P).
effect(omit(G,T),P):-
    occurs(S,omit(G,T),T),complete(S,act(G,X),T),effect(act(G,X),neg(P)),
    not effect(omit(G,T),neg(P)).
effect(omit(G,T),neg(P)):-
    occurs(S,omit(G,T),T),complete(S,act(G,X),T),effect(act(G,X),P),
    not effect(omit(G,T),P),positiveFluent(P).
```

The union of the *event motor* updated to handle omissions with the *planning context* Ctx is called the *action model with omissions* and is denoted by $\mathbb{A}_{Ctx}$.

6.2 Scenarios and Trace

Now that we have defined and illustrated the components of the *action model*, we are interested in its inputs and outputs. As mentioned above, the framework of ethical evaluation takes as input a set of scenarios $\mathbb{S}$ that corresponds to a set of considered actions $\mathcal{A}_c$. We define below the concept of a scenario.

Definition 12 (Scenario)**.** 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) $\mathbb{A}_{Ctx} \cup P$ is consistent and

(ii) $\forall$`performs(`s`,A,T)` $\in P, \mathbb{A}_{Ctx} \cup P \models$`occurs(`$s$`,A,T)`,

where $\models$ denotes the classical skeptical entailment in ASP for stable semantics.

Less formally, a *scenario* is one possible set of performed actions given a planning context, i.e. one possible way the agent has to make a situation evolve. 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* combines the *planning context* and *event motor* to derive 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 13 (Execution trace)**.** Given a *planning context* Ctx, the *execution trace* of a scenario $\sigma = (s, P)$ is defined as $tr_{Ctx}(\sigma) = Th_{\mathcal{L}_{h,o}}(\mathbb{A}_{Ctx} \cup P)$ where $\mathcal{L}_{h,o} =$`holds(`s`,`$\mathcal{F}$`,`$\mathcal{T}$`)`$\cup$`occurs(`s`,`$\mathcal{E}$`,`$\mathcal{T}$`)`.

Informally, the *execution trace* of a scenario is the set of all fluents that hold and all events that occur in the scenario. These are represented as facts in the answer set.

6.2.1 Generating Scenarios

In order to investigate the causal and ethical properties of a given scenario, it is useful to generate all the possible scenarios of a given context. For investigating properties of scenario-based causality as well as certain theories of the Right, this is in fact necessary so as to consider all relevant alternatives and weigh them against one another. This set of simulations can be generated by adding the following lines to $\mathbb{A}_{Ctx}$, with each resulting answer set constituting one scenario.

```
time(1..n).
0{performs(simName,A,T):action(A)}1:-time(T).
```

These rules postulate that the agent must perform 0 or 1 action per time point, exhausting all possible combinations of volitions in the domain. If we are only interested in the properties of certain scenarios but not of all, it is possible to create a predicate which will preclude certain scenarios from further analysis, while still allowing them to participate in the evaluation of others. Such a predicate might for example take the form of `doNotConsider(S):-toAvoid(E),occurs(S,E,T).` Added to the body of the rules of causal and ethical properties, this will preclude the simulations that contain the targeted events.

The steps described so far describe what occurs during the first launch of the program: the *event motor* and *planning context* are launched together to produce multiple answer sets each corresponding to a single scenario, declared in the form of an event trace. At this point, in each scenario/answer set, the simulation variable is called `simName`. For example, if the event `stop` occurs at time 2 in a simulation, the corresponding fact will be `occurs(simName,stop,2)`. At this stage, the only way to distinguish between different scenarios is to recognise that they are in different answer sets. It is only when we clean and edit the resulting answer sets that different simulations acquire unique names that replace the `simName` variable. Specifically, this means that we replace every occurrence of `simName` in the first answer set with `s(1)`, every occurrence of `simName` in the second answer set with `s(2)`, etc[1]. This editing is done through a simple python script that also deletes superfluous information, adds full marks at the end of each line, and generates the two launches of clingo (the ASP grounder and solver). The reason for appealing to this two-step procedure is that it enables us to avoid the main shortcoming that Answer Set Programming shares with other propositional

[1]The numerical order given to simulations is a function of the time the solver took to generate different answer sets, it is therefore arbitrary in ethical terms and can vary from one launch to the next.

techniques, that is, the risk of an exponential slow-down when a large number of objects compose the universe [208]. This script is found in appendix 1.

As demonstrated in this section, the event calculus employed throughout this work is an extension of the original framework whose most notable additions are simulation variables and the handling of omissions. Nevertheless, we did not resort to other possible pervasive extensions of this calculus, such as actions with indirect effects, actions with non-deterministic effects or concurrent actions. This is simply a result of the fact that we did not need to so, so far, nor have we yet set out to intentionally investigate it. We therefore delegate to the future potential extensions in those directions. We also note that we have not formally investigated how far adding simulation variables to the event calculus brings it closer to the situation and fluent calculi, as it is beyond the scope of this work, but we refer to Kowalski and Sadri's work [141] on the logical equivalence of the event and situation calculi as a possible starting point for this endeavour, as well as Thielscher's proposal for a unifying calculus [232]. Another interesting avenue of this type might consist in exploring the formal relation between this form of the event calculus and the DL-MA logic presented in [156], which, though it represents outcomes in a way which does not differentiate between event occurrence and non-occurrence, enables handling actions in a similar way through the use of preconditions.

6.3 Proof of Concept #1 (*collision*)

We illustrate the features and concepts of the *action model* through the following toy example. It is purposefully complex, as it will serve to also illustrate causal and ethical concepts in sections to come.

> (*collision*) Iris is driving her car down the highway, and witnesses a car accident with a very injured driver. She sees that he only stands a chance of surviving if he is brought to safety off the road and gets rapid medical help. They are in an isolated location, so she knows this might not happen. Therefore, she considers the possibility of helping him to die so that he may die a quick and painless death, rather than an agonising one. In addition, if he is not taken off the road, another car will crash into him and kill him. At first, she therefore has three options, bringing him to *safety* behind the guard rail, *killing* him, or doing nothing, where the last two options lead to his death. If she does bring the driver to safety, she has two options: doing nothing, in which case the driver's health will decline and he will die, or performing first *aid* gestures and stabilising his state. In the second case, Iris would now be in a position to perform another action, *calling* for medical assistance. If she does this, then the driver will reach full recovery. If she stops after performing first aid however, the driver will partially recover for lack

of prompt treatment, but this makes a new action available to the driver, muscle *rehab*. If the driver then goes to rehab, he will reach full recovery. If not, he will end up with a limp.

To summarise, there are four actions that Iris is capable of performing at different times, assuming other relevant actions were performed before: *safety*, *kill*, *aid* and *call*. There is one action that the driver is capable of performing if he reaches that point: *rehab*. The outcomes for the driver are full recovery, part recovery, limp and death (by accident, murder or decline). These are automatic events resulting from the action choices of the two agents. The different possible simulations are then:

	t=0	t=1	t=2	t=3	t=4	t=5
s_1	omit(iris)	accident	death	-	-	-
s_2	kill	death	-	-	-	-
s_3	safety	omit(iris)	decline	death	-	-
s_4	safety	aid	omit(iris)	part recovery	omit(driver)	limp
s_5	safety	aid	omit(iris)	part recovery	rehab	full recovery
s_6	safety	aid	call	full recovery	-	-

The different choices available to the agent in the domain create a tree of simulations, which we represent in figure 6.1.

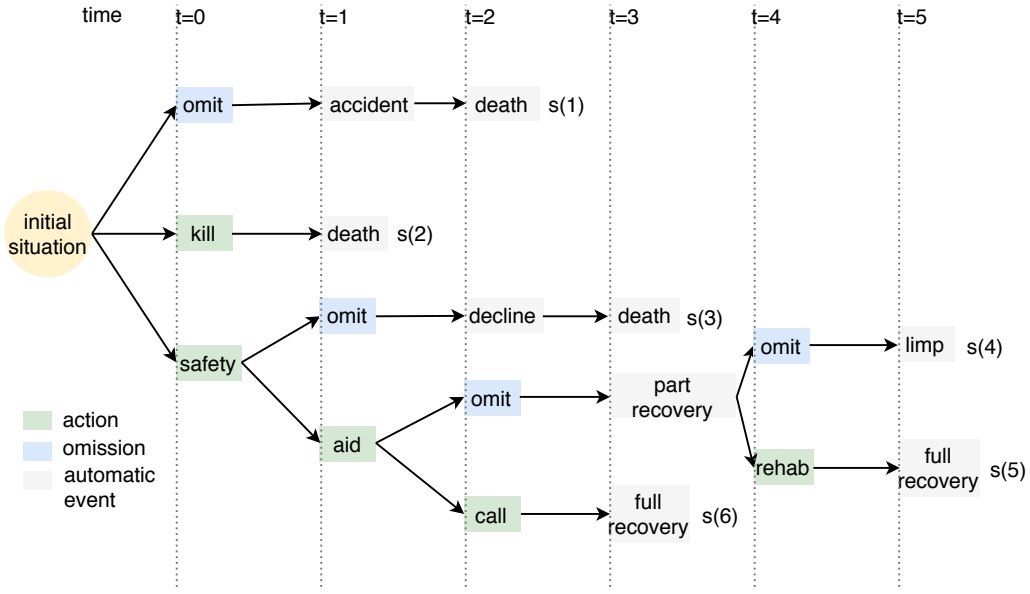

Figure 6.1: Proof of concept: Scenario generation (*collision*)

We now show how it is modelled. The *planing context* is the following.

```
% -------------- Scenario Generation
sim(simName).
0{performs(simName,A,T):action(A)}1:-time(T).
% -------------- Domains
number(-100..100).
time(0..7).
agent(iris;driver).
victim(driver).
positiveFluent(hospital(G);dead(G);hurt(G);
    breathing(G);safe(G);treatment(G);needRehab(G)):-victim(G).
nonInertial(treatment(G);needRehab(G);hospital(G)):-victim(G).
priority(I,U):-volition(I),auto(U).
% -------------- Initial Situation
initially(hurt(driver)).
initially(breathing(driver)).
% -------------- Event Specification
% Act: safety
capable(iris,safety(driver)).
prec(breathing(G2),act(G1,safety(G2))):-agent(G1),victim(G2).
prec(neg(safe(G2)),act(G1,safety(G2))):-agent(G1),victim(G2).
effect(act(G1,safety(G2)),safe(G2)):-agent(G1),victim(G2).
% Act: kill
capable(iris,kill(driver)).
prec(breathing(G2),act(G1,kill(G2))):-agent(G1),victim(G2).
prec(neg(safe(G2)),act(G1,kill(G2))):-agent(G1),victim(G2).
effect(act(G1,kill(G2)),neg(breathing(G2))):-agent(G1),victim(G2).
% Act: aid
capable(iris,aid(driver)).
prec(breathing(G2),act(G1,aid(G2))):-agent(G1),victim(G2).
prec(hurt(G2),act(G1,aid(G2))):-agent(G1),victim(G2).
prec(safe(G2),act(G1,aid(G2))):-agent(G1),victim(G2).
effect(act(G1,aid(G2)),neg(hurt(G2))):-agent(G1),victim(G2).
effect(act(G1,aid(G2)),treatment(G2)):-agent(G1),victim(G2).
% Act: call
capable(iris,call).
prec(treatment(G2),act(G1,call)):-agent(G1),victim(G2).
effect(act(G1,call),hospital(G2)):-agent(G1),victim(G2).
% Auto: accident
```

```
auto(accident(G)):-victim(G).
prec(breathing(G),accident(G)):-victim(G).
prec(neg(safe(G)),accident(G)):-victim(G).
effect(accident(G),neg(breathing(G))):-victim(G).
% Auto: death
auto(death(G)):-victim(G).
prec(neg(dead(G)),death(G)):-victim(G).
prec(neg(breathing(G)),death(G)):-victim(G).
effect(death(G),dead(G)):-victim(G).
% Auto: decline
auto(decline(G)):-victim(G).
prec(breathing(G),decline(G)):-victim(G).
prec(hurt(G),decline(G)):-victim(G).
prec(safe(G),decline(G)):-victim(G).
prec(neg(treatment(G)),decline(G)):-victim(G).
effect(decline(G),neg(breathing(G))):-victim(G).
% Auto: part recovery
auto(partRecovery(G)):-victim(G).
prec(neg(hospital(G)),partRecovery(G)):-victim(G).
prec(treatment(G),partRecovery(G)):-victim(G).
effect(partRecovery(G),needRehab(G)):-victim(G).
% Act: rehab
capable(driver,rehab).
prec(needRehab(G),act(G,rehab)):-victim(G).
effect(act(G,rehab),hospital(G)):-victim(G).
% Auto: full recovery
auto(fullRecovery(G)):-victim(G).
prec(hospital(G),fullRecovery(G)):-victim(G).
% Auto: limp
auto(limp(G)):-victim(G).
prec(needRehab(G),limp(G)):-victim(G).
prec(neg(hospital(G)),limp(G)):-victim(G).
```

Note, we consider that all volitions have priority over all automatic events, so that Iris has time to act before an accident or a deterioration of the driver occurs. Moreover, it should be noted that to simplify the event specification and ensure that events are well defined, we can replace certain specific rules of event preconditions and effects with the general requirement that if a fluent is the effect of an event (that is not an omission), then its negation is part of its preconditions:

```
prec(neg(P),E):-effect(E,P),positiveFluent(P),event(E),not omission(E).
prec(P,E):-effect(E,neg(P)),event(E),not omission(E).
```

Launched with the *event motor*, the above planning context then yields the following result, edited to make each answer set correspond to a simulation name. For questions of space and clarity, we only show the event trace relative to events that occur, but not fluents that hold.

```
sim(s(1)).
occurs(s(1),omit(iris,0),0).
occurs(s(1),accident(driver),1).
occurs(s(1),death(driver),2).
sim(s(2)).
occurs(s(2),act(iris,kill(driver)),0).
occurs(s(2),death(driver),1).
sim(s(3)).
occurs(s(3),act(iris,safety(driver)),0).
occurs(s(3),omit(iris,1),1).
occurs(s(3),decline(driver),2).
occurs(s(3),death(driver),3).
sim(s(4)).
occurs(s(4),act(iris,safety(driver)),0).
occurs(s(4),act(iris,aid(driver)),1).
occurs(s(4),omit(iris,2),2).
occurs(s(4),partRecovery(driver),3).
occurs(s(4),omit(driver,4),4).
occurs(s(4),limp(driver),5).
sim(s(5)).
occurs(s(5),act(iris,safety(driver)),0).
occurs(s(5),act(iris,aid(driver)),1).
occurs(s(5),omit(iris,2),2).
occurs(s(5),partRecovery(driver),3).
occurs(s(5),act(driver,rehab),4).
occurs(s(5),fullRecovery(driver),5).
sim(s(6)).
occurs(s(6),act(iris,safety(driver)),0).
occurs(s(6),act(iris,aid(driver)),1).
occurs(s(6),act(iris,call),2).
occurs(s(6),fullRecovery(driver),3).
```

As such, we see how the domain contains actions, omissions, fluents (inertial and non-inertial), different simulations, and priorities between events. We also see that different actions are available to different agents (e.g. *aid* to Iris, *rehab* to the driver), that different actions might be available at the same time (e.g. *safety* and *kill*), that choosing some volitions precludes doing others (omitting to act at t=0 precludes all other actions, *aid* makes *call* possible, etc.). In the next chapter, we present the *causal model*.

Part IV

Causal Model

Chapter 7

Contribution: A Causal Model

7.1 Characterising Approaches to Causation

7.1.1 Counterfactual Causation

Going back to Hume [121], the notion of causality has been widely analysed and defined, appraised variously through counterfactual, probabilistic or structural models, with the aim of reducing propositions of the form "E1 is a cause of E2" to more primitive concepts (e.g. [51, 147, 148, 184, 187]). Since the present model does not address issues linked to statistical causal search, rational learning or uncertainty about event effects, it is counterfactual approaches that are of most interest here.

Logicians have elaborated numerous counterfactual analyses of causation, most famously by appealing to the idea of a "nearest possible world" [229] or sets of such worlds [148]. The claim is that a counterfactual 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" [169], and some have argued that judgements about the closeness of possible worlds are made on the basis of beliefs about causal laws rather than the other way around.

Some accounts of causation are also purely counterfactual, such that they postulate definitions of the from: *E1 causes E2 if and only if, had E1 not occurred, E2 would not have occurred either.* This idea corresponds to the legal concept of a "but-for" test: but for A, B would not have occurred. John Mackie's INUS condition, for example, is a slightly more complex example of such accounts, which postulates that *E1 causes E2 if and only if E1 is a(n insufficient but) necessary part of a(n unnecessary but) sufficient condition for E2* [160]. Other works include [87, 99, 101, 113, 249, 250]. But even counterfactual theories of causation that do not appeal to possible worlds run into a

number of issues. To begin, they usually cannot distinguish between direct and indirect causation or handle the asymmetry of causation. An event is said to directly cause another event if its influence is not mediated by another event, and indirect otherwise [228], while causation is asymmetrical in that if E1 causes E2, then, typically, E2 will not also cause E1. Yet a counterfactual test has no way of expressing this. In addition, two important challenges to such accounts come from the problem of *preemption* and the problem of *over-determination*. Preemption occurs when when one cause can be replaced by another to produce an effect, and over-determination occurs when there are more causes than are necessary to produce the effect. Consider the following case of preemption from [138]:

> (*Suzy-Billy*) Suzy throws a rock at a bottle (s-throws), and shatters it (shatters). Billy was standing by with a second rock. Had Suzy not thrown her rock, Billy would have shattered the bottle by throwing his rock (b-throws).

Here, it is not the case that if *s-throws* had not happened, then *shatters* would not have happened, since *b-throws* would have made *shatters* happen. Likewise, over-determination would have occurred in the case that Suzy and Billy both threw rocks at the same time and broke the bottle, assuming the bottle would have shattered even if only one of them had thrown a rock. Naive counterfactual accounts would consider that in the first case Suzy's throw is not a cause of the bottle shattering since it would have shattered even if she hadn't acted. In the second case, they would consider that neither Suzy nor Billy are causes, for the same reason. Yet in both cases their responsibility is surely engaged. An affector can cause an end-state without the latter being counterfactually dependent on the former.

For the reasons sketched out here, we take the stance of defining causation in a primarily non counterfactual way by first employing mechanisms of *event-based causality* (i.e. in terms of pre-conditions) and then using counterfactual and conditional tests to buttress these relations, rather than directly appealing to counterfactual definitions. This is possible because the state transition system at the centre of the event calculus effectively affords us with an existing causal structure, whose semantics allow us to avoid the difficulties faced by primarily counterfactual accounts, such as handling cases of preemption and over-determination.

7.1.2 Scalar Causation

With some notable exceptions, most models of causation treat it as an all-or-nothing concept. 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 in people's reactions [188, 253]. Two recent formal models do show a

sensitivity to this issue. In [45], Chockler and Halpern give a definition of shared responsibility that is responsive to properties of diffusion across a group in which many agents participate in an outcome, but where no individual agent is necessary for it. In other words, they tackle cases where no single agent can alter a given outcome by performing a different action. Considering the outcome of a vote and k the number of changes needed to make one vote critically impacting, they propose that an agent's degree of responsibility for an outcome is equal to 1/(1+k). For example, given a 11/10 outcome to a vote in which 21 agents participated, each one of the 11 voters who voted for the successful outcome are considered to have a degree of responsibility of 1, since all their votes were necessary to make their choice successful. Reversely, if the same vote had had an outcome of 21/0, then all agents would have had a degree of responsibility of 1/11 because, in order for any one vote to have a critical impact on the outcome, 10 votes would have had to be made in the other direction. The authors also model epistemic states to express the relationship between degree of responsibility and degree of blame for an outcome.

Likewise, Van de Poel *et al.* philosophically investigated of the *problem of many hands*, which pertains to the challenge of assigning responsibility and blame to agents whose role is individually insufficient but communally sufficient for producing a given outcome [241]. This is pertinent to such things as over-fishing or breaching the carbon-emission thresholds in a given area. This problem is also a challenge in the aftermath of disasters, such as the Deepwater Horizon oil spill, as these are often a product of the interaction between the effects of a great number of actions and actors. Appealing to a variant of the coalition epistemic dynamic logic (CEDL), the authors address responsibility from both a prospective direction (pertaining to obligation or virtue) and retrospective direction (pertaining to accountability, blameworthiness or liability). This framework however cannot handle notions of counterfactuals, nor can it tackle the (*Suzy-Billy*) case. In addition, these kinds of work focus on shared responsibility within groups of agents, but do not address degrees of responsibility dependent on other factors, such as the nature of the events composing a causal relation or the presence of other causal relations in a domain. Yet these might be determining, as we discuss now.

Nature of the Events

The fact that an action rather than an omission led to a given outcome might make us morally appraise an agent's choice differently. Likewise, an agent's answerability is usually higher in the case that the agent acts in such a way that an automatic event occurs than if the agent acts in such a way that another agent's action becomes possible. We might also appraise differently the fact of making an outcome occur and the fact of stopping an outcome from occurring. These observations have therefore led us to produce an *event-based* model of causality which distinguishes four initial

types of causal relations contingent on the nature of the entities that compose them, as volitions or automatics events, and as produced events or avoided ones. These direct causal relations are then combined to express transitive connections between events that might be linked through a number of other events.

Context

When we say that an agent's has led to a given outcome, and we aim to ascribe responsibility on that basis, it might also be compelling (or necessary) to know such things as whether the outcome could have been avoided at all, or been produced by other means. The work in the second part of this section seeks to address this, not by giving a formal and static definition of moral responsibility, but by providing a framework for modelling the properties that might make up such a definition. As such, after modelling *event-based causality*, we turn to *scenario-based causality* and derive four properties of interest by exploring alternative versions of an original scenario. We account for simple counterfactual 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?" These properties scrutinise and buttress the causal relations identified in the first stage by helping to decipher, strengthen or diminish the attribution of agent responsibility.

7.2 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") [100]. We here focus on the second notion, while the information given by the *event specifications* in the *action model* hint at definitions of type-causality. 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 [108, 225]. 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. We do the latter, and here distinguish different facets of such actual causality.

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. Yet works in computational ethics typically

neglect this aspect, and fail to address prevention distinctly from causation. 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 summarised in table 7.1, **causes** denotes strong supporting causality, **prevents** denotes strong opposing causality, **enables** denotes weak supporting causality, and **excludes** denotes weak opposing causality. As such, 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. Throughout, we call the event at the beginning of the causal chain the *affector* and the event at the end of the causal chain the *end-state*. The nature of the affector 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 end-state.

Table 7.1: Matrix of causal relations

	Supporting	Opposing
Strong	causes	prevents
Weak	enables	excludes

7.2.1 Supporting Causality

The process of defining causality based on the event calculus amounts to inferring causal relationships from material implication. In other words, we consider *E1 causes E2* to mean that, given the state of the world, "if E1 is executed, then E2 will also be executed" *and* that "the execution of E1 causes E2."

Definition 14 (Causing)**.** 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 15 (Enabling)**.** 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 discuss the transitive powers of other relations in a specific section (7.2.5), 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 model causing in the following way. `r(S,causes,E,T,U)` indicates that event E, which happened at time T in simulation S, causes the automatic event U. In addition, for the sake of simplifying rules that pertain to causal chains, we consider that causing is reflexive such that all events that occur also cause themselves. The referenced time point denotes the time at which occurred the first event within a causal chain. Supporting causal relations are denoted by the `supRel` predicate.

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

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 actions that are 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(G,T)):-
    holds(S,F,T),prec(F,act(G,X)),complete(S,act(G,X),T),occurs(S,omit(G,T),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.

7.2.2 Opposing Causality

Definition 16 (Preventing)**.** 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 17 (Excluding)**.** 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

In order to model these relations, we first 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.

`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.

`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. F can be a positive or negative fluent, so that if F has

the form P, then terminating it will means making neg(P) true, and if F has the form neg(P), then terminating it will mean making P true. It is important to handle both cases since the preconditions to events can take the form of positive or negative fluents alike. Just like events can be caused by negative fluents that have been made true, so can they be prevented by positive fluents that have been made true. The purpose of this predicate is also to allow for indirect cases where E affects a non-contiguous fluent. It corresponds to the (a) clause of the definition of *prevents* and *excludes*.

`relevant(S,E1,T,E2)`, through the `irrelevant(S,E1,T1,E2,T2)` predicate, identifies 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. `didHold(S,F,E,T)` serves to properly handle the role of non-inertial fluents in this process. Because an event occurrence takes up one time point, and because non-inertial fluents are true for one time point, if a precondition to an end-state was true at the time of occurrence of the affector event, it will not be true after it. Yet we still want to consider that, had the affector event not occurred, the precondition was fulfilled.

```
hyp(F1,F2):-prec(F1,U),auto(U),effect(U,F2).
hyp(F1,F3):-hyp(F1,F2),hyp(F2,F3).
canArise(S,U):-auto(U),occurs(S,U,T).
canArise(S,I):-volition(I),complete(S,I,T).
transTerm(S,E,P,T):-occurs(S,E,T),effect(E,neg(P)),holds(S,neg(P),T+1).
transTerm(S,E,neg(P),T):-occurs(S,E,T),effect(E,P),holds(S,P,T+1),positiveFluent(P).
transTerm(S,E,F2,T):-transTerm(S,E,F1,T),hyp(F1,F2).
didHold(S,F,E,T):-holds(S,F,T),prec(F,E),nonInertial(F).
irrelevant(S,E1,T1,E2,T2):-transTerm(S,E1,F1,T1),prec(F1,E2),prec(F2,E2),
    not transTerm(S,E1,F2,T1),not holds(S,F2,T2),T1<T2,time(T2),not didHold(S,F2,E2,T1).
relevant(S,E1,T1,E2):-
    transTerm(S,E1,F1,T1),prec(F1,E2),not irrelevant(S,E1,T1,E2,T2),T1<T2,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.

```
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),not omission(E),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),not omission(E),action(A).
```

As seen in the above rules, we preclude omissions from being considered as affectors in these definitions of preventing and excluding. We explain why in the following section.

7.2.3 The Causal Properties of Omissions

In terms of *event motor* dynamics, an omission makes true what available actions would have made false and false what they would have made true. However, in terms of causal traces, we do not take into account what omission are said to make false, nor the events that could be considered to have been prevented as a result. The reason we do not integrate this information is that it would state as true something that is in fact uncertain: these terminated fluents were potentially not going to be initiated at all, and the corresponding events not about to occur at all. Indeed, among the many actions that might have been available to the agent at the time of its omission, only one of them would actually have been performed at that time, since only one action can ever be performed at each point in time. Therefore, only the effects of that particular action should be considered when considering the effects of the omission, so that we do not consider that we prevented events that wouldn't actually have been caused. Yet we have no way of knowing which action is the relevant one. Moreover, even if we had a model in which multiple actions could be performed simultaneously, this issue would still arise in cases where different actions are mutually exclusive in the domain. For example, we can imagine a train on tracks that comes to a triple intersection: the driver can go left, right, or do nothing and let the train continue straight ahead (see figure 7.1). If he does nothing, what did he prevent? The train running on the left tracks or the right? Something like *a little bit of both but neither completely*, since going left and going right are mutually exclusive and could not have both occurred. This, then, is why we avoid considering that an omission has the power of direct prevention and exclusion, even though in cases where only one action is available no uncertainty would arise.

Such uncertainty does not arise when considering actions because they terminate fluents that *were already true*, rather than fluents that *could have been made true*. This shows an asymmetry between actions and omissions: actions prevent existing threats, omissions do not. Omissions cause existing threats, actions do not (relative to omissions, causing is akin to preserving). In the triple intersection case just mentioned, this asymmetry materialises in the fact that turning left or right *prevents* what

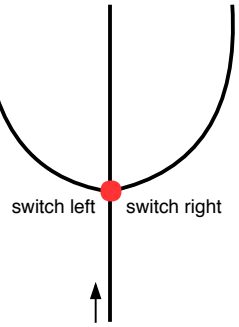

Figure 7.1: Mutually exclusive actions

the train threatened to do, continuing to go straight. Reversely, omitting to act would *cause* this threat to materialise. As such, modelling actions and omissions makes salient the fact that the notion of *preexisting threats* is central to understanding the ontological difference between these two types of volitions.

Omissions can nevertheless already be found to prevent and exclude some events through the transitivity of causing. If an omission causes an event that prevents a third event, the omission prevents that event too. This will concern events that are down the causal chain of the omission, and not events that are in the hypothetical neighbour chains that other available actions would have produced. For example, take the case of a drowning child, whom I omit to save. My omission *causes* the death, which in turn *prevents* the child's parents from being happy. My omission therefore *prevents* this happiness too, as this end-state is down the causal chain from my omission. Consider furthermore that at the time of my omission, three actions were immediately available to me: saving the child (preventing the death), throwing another child in the water (causing a death), or jumping in myself (causing a death – I cannot swim). If indeed my omission causes the death of the first child, it seems inappropriate to argue that I prevented the other child's death as well as my own. If this was the case and we compared the number of lives saved to lives taken by my omission, it would get a faulty score of +1. This is because these events are in hypothetical parallel causal chains, and not downhill from my omission. To summarise, if an action prevents an event, then omitting to act instead causes the event, but if an action causes an event, omitting to act instead *does not* prevent the event. We now turn to discussing further causal rules that will, among other things, enable the framework to assign further and suitable causal powers to omissions.

7.2.4 Choosing a Volition

When an agent makes a choice about performing a certain action or refraining to act altogether, it is important to be able to reason over the fact that previously available options might be made unavailable by that choice. We therefore state that, if the agent chooses a volition (performed action or occurred omission), then any other volition that was complete at the same time but not chosen is considered *excluded* by the chosen volition, unless it becomes complete again at a later time (in which case the agent has not excluded it as it remains available to him). This requirement for exclusion is slightly different from the definition discussed above, since there is no reasoning over the preconditions of events, but it is similar in that it aims to trace what choices for acting the agent has throughout the simulation.

```
r(S,excludes,I1,T1,I2):-occurs(S,I1,T1),complete(S,I2,T1),not complete(S,I2,T2),
    volition(I1;I2),T1<T2,time(T2),I1!=I2.
```

As such, at this point, the agent is able to reason about excluded omissions and their effects, and about excluded actions, but *not* about the potential effects of excluded actions. When choosing a volition, the agent excludes all other volitions, prevents the effects that the *omission* would have had, but is yet to be able to reason about the effects that the other *actions* would have had, had they been performed. Figure 7.2 summarises the opposing causal relations that can presently be inferred relative to picking a volition.

Each red arrow is drawn assuming that the volition it stems from has been chosen by the agent. As such, the different arrows belong to different simulations within the five that are represented. For example, the omission at t=0 excludes both other actions that were complete at the same time in the two simulations in which it occurs (s(1) and s(2)). We can also note by looking at this figure that nothing is said about the omission at t=2 in s(1). In particular, it is not considered excluded by the actions available at t=0. This is because omissions to do not have preconditions *per se* but depend on the possibility of acting, hence they are not traceable in such a way. We will attend to this in the section on the transitivity of causal relations, which we turn to now.

7.2.5 Transitivity

The transitive powers of causing are straightforward: if I cause something that causes something else, then it is easy to see I cause the final event. If I push a glass off a table, and the glass shatters and the water spills and wets the carpet, I caused all of these things. Likewise, if I cause an event that prevents (or enables, or excludes) another event, then I prevent (or enable, or exclude) that final event. The transitive powers of enabling, preventing and excluding are less simple to

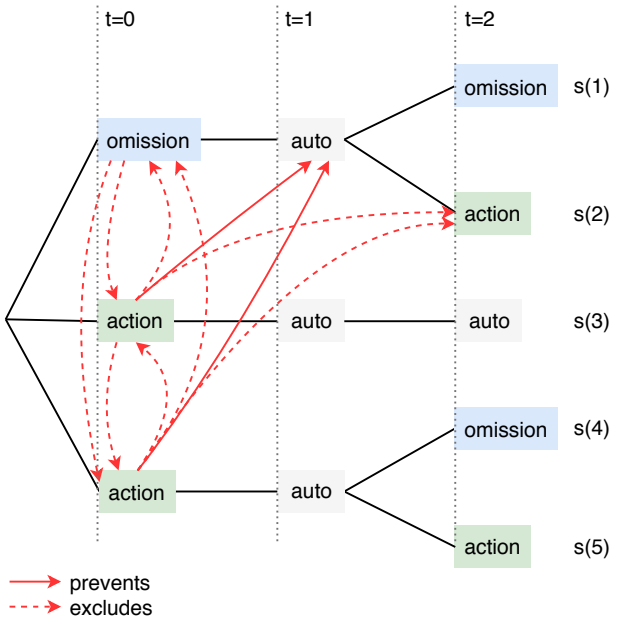

Figure 7.2: Opposing relations + excluded volitions

envision, because instead of requiring factual reasoning about automatic events, they imply more complex reasoning about both automatic events and volitions, in potentially counterfactual ways. The challenge in this section is then to complete table 7.2.

Table 7.2: Incomplete transitivity of causal relations

Chain	**Transitivity**
causes(E1,E2),causes(E2,E3)	causes(E1,E3)
causes(E1,E2),enables(E2,E3)	enables(E1,E3)
causes(E1,E2),prevents(E2,E3)	prevents(E1,E3)
causes(E1,E2),excludes(E2,E3)	excludes(E1,E3)
enables(E1,E2),causes(E2,E3)	?
enables(E1,E2),enables(E2,E3)	?
enables(E1,E2),prevents(E2,E3)	?
enables(E1,E2),excludes(E2,E3)	?
prevents(E1,E2),could cause(E2,E3)	?
prevents(E1,E2),could enable(E2,E3)	?
prevents(E1,E2),could prevent(E2,E3)	?
prevents(E1,E2),could exclude(E2,E3)	?
excludes(E1,E2),could cause(E2,E3)	?
excludes(E1,E2),could enable(E2,E3)	?
excludes(E1,E2),could prevent(E2,E3)	?
excludes(E1,E2),could exclude(E2,E3)	?

Transitivity for Enabling We first consider the transitive powers of enabling relative to supporting relations, that is, causing and enabling. Take the example given in section 7.2: John tells Pat where Sam is and Pat harms Sam, so John enables Pat's action, while Pat causes the harm. What of John's relation with the harm done? Does he enable it too? Take another case: I lend my car to a friend of mine so that he may drive a friend of his to a bank where this last person will commit a robbery. Here, by lending my car I enable my friend's action to drive, which enables his friend's action of robbing the bank. What is my relation to the robbery?

In these cases, the affector does not make the preconditions to the end-state true, rather, *it makes it possible for them to become true*, given that some other relevant affector volition also occurs. In some cases, this may not be morally relevant, and the agent undertaking the first volition might be as morally responsible at the agent undertaking the second volition that is directly linked to the end-state. For example, if I give my ID to my underage little brother and he buys alcohol for his friends, and one of them passes out from the alcohol, it would seem I am fully responsible even if I didn't directly make true the preconditions for this passing out, since I didn't hand out any alcohol myself. In other cases, the removal may be morally significant. In the robbery case, even if it shown that I knew what my car was going to be used for, it is likely that I will get a lesser conviction than my driver friend. Based on this and the conclusion that the distinction is morally significant in at least some cases, we chose to distinguish through a different predicate cases of direct enabling from cases of transitive enabling, which we call `helps(enables)`. If a volition enables a volition which causes or enables a third event, then the volition *helps to enable* that event. We do not define a "helps to cause" predicate because in both cases volitions are involved, meaning that we deal not just with automatic events but also with the choices of agents, so it is appropriate to appeal to a weaker causal bond than one of straightforward causation. Throughout, in fact, relations of weak causality trump relations of strong causality, since volitions are dominant over what we can consider to be recessive automatic events: if a volition is involved in a relation (except as affector), then the relation will necessarily be one of weak causality, but if an automatic event is involved in a relation, it is not the case that the relation will necessarily be one of strong causality. A relation will be one of strong causality if events involved are *exclusively* automatic (except, again, for the affector which may be any kind of event).

Relative to opposing relations, we define the `helps(excludes)` predicate. A volition which enables a volition which prevents or excludes a third event *helps to exclude* that event. We can envision such a scenario: instead of being a bank robber, my friend's friend is a fugitive, and my lending of the car enables my friend to drive the fugitive to a safe-house, excluding the police's action of arresting him. Here, I help to exclude that possibility. But this predicate also identifies cases that are somewhat counter-intuitive. It is conceivable that I help to exclude an end-state that my

original volition in fact originally made possible. In other words, if my volition hadn't occurred, the end-state would never even have been possible, and therefore there would have been no need and no possibility to exclude or prevent it. This is often the case in simple train tracks examples. Imagine that I am driving a train and notice on the tracks to my left that someone is walking. I am a malicious driver and want to scare that person, so I switch the train onto those left tracks. I do not want to kill them, however, so just in time I switch back onto the tracks where there is no one. Here, I help to exclude the crash, since my first action of switching left enables my second action of switching right and preventing the crash. This, in the ethical framework, will be considered a morally valuable thing. Yet we hardly want to say that my first action was morally good, it is at best neutral. Other rules of transitivity also occasionally produce counter-intuitive results. This is the case in some instances of preventing where the affector causes an event which prevents a third event. Consider a train which crashes into a cow and stops as a result. There was another cow further down the tracks. The fact that the train was running forwards caused the first crash which then prevented the second crash. As such, the fact that the train was running forwards prevented the crash with the second cow, yet it is also what made it possible in the first place.

The way we will be able to distinguish these cases from genuine cases of helping to exclude or preventing will be to test whether the relationship between the affector volition and the end-state is counterfactually valid. In genuine cases, if the volition does not occur, then the end-state will not occur either. In other cases such as the malicious driver case, the end-state will still occur: even though I help to exclude the crash, had I not performed my first volition (switching left), the crash would still not have occurred - the end-state is the same regardless of that volition, so the relation is invalid in a counterfactual sense. We will model and discuss this property in section 7.3.1. Predicates capturing the transitive powers of enabling are the following, categorised as relations of transitive supporting causality and transitive opposing causality, or `supRelTrans` and `oppRelTrans`.

```
supRelTrans(helps(enables)).
r(S,helps(enables),E1,T1,E3):-r(S,enables,E1,T1,E2),r(S,R,E2,T2,E3),
    supRel(R),event(E1;E3),not r(S,causes;enables,E1,T1,E3),T2>T1.
oppRelTrans(helps(excludes)).
r(S,helps(excludes),E1,T1,E3):-r(S,enables,E1,T1,E2),r(S,R,E2,T2,E3),
    oppRel(R),event(E1;E3),not r(S,prevents;excludes,E1,T1,E3),T2>T1.
```

The `not r(S,causes;enables,E1,T1,E3)` and `not r(S,prevents;excludes,E1,T1,E3)` clauses in the above rules serve to avoid the redundancy that can arise through the causal powers of omissions. Because they do not change the state of the world, omissions can cause and enable events that are also caused and enabled by events that precede them (this is not the case for actions, if an

action causes or enables another event, the events preceding the action will only have a transitive relationship to that last event, since the action breaks the direct link). As such, if action A causes event E with an omission O occurring in between, it will also be true that A enables O and that O causes E. Without the clauses discussed, the rules of transitivity would therefore consider that A *helps to enable* E, alongside the fact that A *causes* E, which would be redundant. The same is true for helping to exclude.

Transitivity for Preventing and Excluding We first consider the transitive powers of preventing and excluding relative to opposing relations, that is, preventing and excluding. Rules of transitivity between opposing relations are crucial for getting the full picture of a situation. There are outcomes that the agent does not cause, enable, prevent or exclude, but for which it is still responsible in so far as it could have acted in such a way that these events did not occur. An example.

> My friend and I know that a bomb has been planted in a building, though we did not plant it ourselves. My friend does not want it to go off and wants to call the police to warn them so that they can disarm it, but I steal his phone because I do want the explosion to happen. I exclude his attempt to prevent the explosion.

In our model, unless we appeal to opposing x opposing transitivity rules, the framework has no way of assigning any responsibility to me for participating in the explosion, yet morally I am evidently to blame somewhat. But transitivity investigation for opposing relations poses a novel problem: it requires counterfactual reasoning. Indeed, if I want to investigate the relationship between my action A1 that excluded my friend's action A2 and an event E which would have been prevented by the occurrence of A2, then I must have a way of modelling the simulation in which A2 does occur. This simulation is contrary to fact, the reasoning process is counterfactual. A number of questionings and complications arise from this. To discuss them, we now expose the rules that we modelled, then unpack them.

```
supRelTrans(impedes(prevents;excludes)).
r(S1,impedes(prevents),E1,T1,E3):-
    r(S1,prevents,E1,T1,E2),r(S2,prevents,E2,T2,E3),occurs(S1,E3,T3),
    {r(S1,R,E1,T1,E3):rel(R)}0,sameHistory(T1,S1,S2),T1<=T2,T1<T3,E1!=E3.
r(S1,impedes(excludes),E1,T1,E3):-
    r(S1,prevents,E1,T1,E2),r(S2,excludes,E2,T2,E3),occurs(S1,E3,T3),
    {r(S1,R,E1,T1,E3):rel(R)}0,sameHistory(T1,S1,S2),T1<=T2,T1<T3,E1!=E3.
r(S1,impedes(excludes),E1,T1,E3):-
```

```
    r(S1,excludes,E1,T1,E2),r(S2,prevents,E2,T2,E3),occurs(S1,E3,T3),
    {r(S1,R,E1,T1,E3):rel(R)}0,sameHistory(T1,S1,S2),T1<=T2,T1<T3,E1!=E3.
r(S1,impedes(excludes),E1,T1,E3):-
    r(S1,excludes,E1,T1,E2),r(S2,excludes,E2,T2,E3),occurs(S1,E3,T3),
    {r(S1,R,E1,T1,E3):rel(R)}0,sameHistory(T1,S1,S2),T1<=T2,T1<T3,E1!=E3.
```

`r(S1,R1,E1,T1,E2),r(S2,R2,E2,T2,E3)` First, and most ambiguous, is the requirement to establish what exactly we mean by wanting to investigate the relation between "my action A1 that excluded my friend's action A2 and an event E *which would have been prevented by the occurrence of A2*". Do we mean that A2 would have prevented E in all possible cases (i.e. in all simulations in which it occurs)? Or just in some of them? Or just in a specific one? Taking the explosion example, are we interested in the case in which, had my friend kept his phone, *there was no way* the explosion would have happened, or case in which *there was a relevant scenario* in which the explosion did not happen, or a case in which *there was at least some chance* that the explosion did not happen? Is my role in the matter more interesting or relevant in one or the other of these options? We can rephrase the question in respective terms of, if A1 hadn't excluded A2, then A2 *would necessarily* have prevented E; A2 *would, assuming other things were also true,* have prevented E; A2 *could* have prevented E2. Arguably, all these cases are interesting, and all could be modelled. Nevertheless, for concision and clarity, and because we consider that responsibility is engaged from the moment an end-state *could* have been modified, we only choose to model the last case. In other words, we consider that an event E1 has a supportive relationship to an event E3 if E1 prevents/excludes an event E2, and there exists a simulation in which E2 occurs and prevents/excludes E3. We call these resulting supportive relations *impeding preventing* and *impeding excluding*, depending on whether they exclusively concern automatic events or also concern volitions.

`occurs(S1,E3,T3)` Transitivity rules between any two kinds of opposing relations will yield supporting relations; implicit in this fact is that only end-states that occur in the simulation are of interest. Indeed, if in simulation S1, A1 excludes A2 which could have prevented E (in a simulation S2), but E does not actually occur in S1, then there is not much to gain from modelling these rules, since whether I do or don't perform A, E does not occur, so that A cannot be considered a relevant determinant of E. If the explosion did not go off, then it is unimportant that I stole my friend's phone (except in terms of intentions, but we do not touch upon these here). As such, we specify that these rules of transitivity must hold between an affector and an end-state that occurs within the simulation.

`sameHistory(T1,S1,S2)` Finally, we discuss an issue that will be recurrent in the upcoming sections. This is the "same history problem." This problem stems for the complexity that is created by having to handle a combination of multiple volitions and multiple simulations within a single

framework. Broadly speaking, the issue is about selecting relevant simulations for analysis. When we want to test whether there is a simulation in which (a prevented/excluded) E1 prevents/excludes E2, we do not necessarily want to include all simulations in which E1 occurs. Indeed, it is possible that there are simulations in which E1 occurs but which are irrelevant because the events that occurred in them at earlier time points differ from those that occurred in the simulation in which we investigate the causal relation. For example, in the explosion scenario, we want to find out whether, if my friend had access to his phone, it would have been possible for him to prevent the explosion. Let's consider that as it is, he actually could not have stopped the explosion because the timer on the bomb was too short and it would have went off before the police had time to disarm it. Hence in the model, the result we want to find is that there is no supportive transitive relation between my action of stealing the phone and the bomb going off, since whether or not I stole the phone, the bomb would have gone off (again, we are not concerned with intentions here, but solely focus on actual causality). If we now imagine that earlier in the scenario, the bomber agent had a choice between setting a long time window and a short time window for the bomb to go off, then there exist in the set of simulations a simulation in which the police could have had enough time to stop the explosion. We do not, relative to the present simulation at the time of my stealing action, want to consider this a relevant simulation, since, from where we "stand", it is not a possible alternative scenario. It is not a valid alternative because its past is different from the past of our considered simulation: in the present simulation, the the bomber agent set a short time window. Without the `sameHistory` predicate, the model would have no way of taking this into account and would therefore give a false positive by determining that my action impedes the excluding of the explosion when in fact it did not, by finding a simulation in which the bomb does not go off. We define this predicate by stating that two simulations have the same history up to a time T if all events that occurred in one also occurred in the other up to but not including T.

Transitive and non-transitive relations can overlap: it may be the case that A causes E but also prevents an event that would have prevented it. We argue that it is informative to know whether more than one relation holds. However, if we were to deem it redundant, and assuming we gave precedence to non-transitive relations, we could preclude multiples by adding the formula

to the right side of the rules of transitivity. The form `{xx:yy}0` ensures that the rule fails if an xx such that yy is found.

```
differentHistory(T2,S1,S2):-
    occurs(S1,E,T1),not occurs(S2,E,T1),sim(S2),event(E),time(T2),T1<T2.
```

```
differentHistory(T,S2,S1):-differentHistory(T,S1,S2).
sameHistory(T,S1,S2):-not differentHistory(T,S1,S2),time(T),sim(S1;S2).
```

We now turn to the transitive powers of preventing and excluding relative to supporting relations, which yield opposing relations. We discuss them separately for preventing and excluding because in this case they do not have equivalent ramifications. Relative to events that exclude an event that could have caused or enabled a third event, we model the corresponding rules in the same way as above, namely by ensuring that the same history problem is handled and that the hypothetical is a "could" rather than a "would." The only difference with the previous rules is that we naturally specify that the end-state *does not arise*, rather than *does occur*. Indeed, if I exclude an event that would have caused another event, then the relation is only meaningful (and meaningfully *opposing*) if the final event does not actually arise in the simulation. As in the definitions of prevents and excludes in section 7.2.2, we use the `canArise` predicate which identifies the occurrence of automatic events and the fact that volitions are complete and available to the agent. The resulting relation is called *impeding enabling*.

```
oppRelTrans(impedes(enables)).
r(S1,impedes(enables),E1,T1,E3):-r(S1,excludes,E1,T1,E2),r(S2,causes,E2,T2,E3),
    not canArise(S1,E3),sameHistory(T1,S1,S2),T1<=T2,event(E3).
r(S1,impedes(enables),E1,T1,E3):-r(S1,excludes,E1,T1,E2),r(S2,enables,E2,T2,E3),
    not canArise(S1,E3),sameHistory(T1,S1,S2),T1<=T2,event(E3).
```

Relative to preventing, having rules that identify when an agent prevents an event that would have caused or enabled another event in fact turns out to be of little use, because they respectively overlap with the non-transitive definitions of preventing and excluding. Indeed, these are simply cases in which the affector terminates a precondition to the end-state somewhere along the causal chain that links them. As such, there is no need to model them. There exists one exception to this which comes from the way we modelled omissions. Indeed, as discussed previously, though the model so far has the capacity to infer when actions have been excluded, it cannot do so for omissions that would have been possible far down causal chains, since omissions do not have specific preconditions that can be seen to be terminated. The only method for excluding actions that has been discussed so far is in the Choosing a Volition section, whereby when an action is performed the omission at the same time is excluded. Omissions that could have been possible at later times down the causal path of these omissions are not handled. But the concept of preventing an event which enables another event can allow us to handle the existence of these excluded omissions without appealing to preconditions. This, on the scale of the framework, is a matter of detail, but we still make the choice of using this rule to fill in the gap. We therefore consider that an event which prevents an

event which could have enabled a third event *excludes* that event. We do not, however, model the relation between an event that prevents an event that would have caused another event, since, as discussed, this is fully subsumed within the definition of preventing.

```
r(S1,excludes,E1,T1,E3):-r(S1,prevents,E1,T1,E2),r(S2,enables,E2,T2,E3),T1<=T2,
    not canArise(S1,E3),sameHistory(T1,S1,S2),event(E3).
```

Figure 7.3 illustrates what has been added to the framework since figure 7.2, showing the new opposing relations that can be inferred from a set of simulations when a choice of volition is made. Notably, when an action is chosen, this action

(a) excludes all other available volitions,

(b) prevents the effects of the omission if they are automatic events,

(c) excludes the effects of the omission if they are volitions, and

(d) impedes the enabling of the effects of the other actions.

When an omission is chosen, it

(a) excludes all other available actions and

(b) impedes the enabling of their effects.

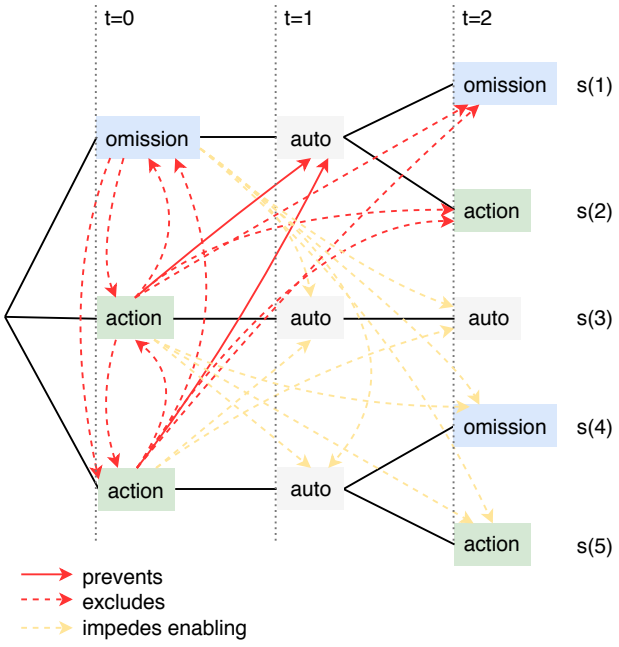

Figure 7.3: Opposing relations + excluded volitions + transitive opposing relations

Meta-transitivity The rules of transitivity defined above add the capacity to handle one additional layer of events within causal chains, but stop there. We can reason over event E1 which prevents event E2 which could have prevented event E3, but cannot say anything about the relationship between E1 and a potential E4 which could have been prevented by E3. This is unlike the transparent rule of transitivity for causing which can be infinite: I can cause (or enable, prevent, exclude) an end-state through any number of automatic events. Relative to opposing relations, we satisfy ourselves with these somewhat limiting rules of transitivity for the reason that further rules would imply having to reason counterfactually about counterfactuals, effectively modelling a complex succession of "coulds." The moral meaning of the resulting relations becomes difficult to grasp. Relative to supporting relations, however, we postulate a final rule of supporting "meta-transitivity" that enables iterations over an infinite number of events leading to any relation of transitivity. We state that an event E1 which *enables* an event E2 which has a relation of transitivity with a third event E3 itself assumes that transitive relationship with E3. We make enabling transparently transitive with regard to transitive relations. For example, if my action A1 enables A2 which helps to enable A3, then A1 helps to enable A3. If my action A1 enables A2 which impedes the preventing of E, then A1 impedes the preventing of E. Taking the explosion example from above, let's consider a third person who helped me stealing my friends phone by creating a necessary diversion, and that the explosion did occur. This third person enabled my sealing action which impeded the prevention of the explosion. We then consider that they, too, impeded the prevention of the explosion. We do not need to specify an equivalent rule for causing since its rule of transitivity already captures these links.

```
relTrans(R):-supRelTrans(R).
relTrans(R):-oppRelTrans(R).
r(S,R1,E1,T1,E3):-r(S,enables,E1,T1,E2),r(S,R1,E2,T2,E3),relTrans(R1),event(E1;E3),T1<T2.
```

To summarise, we have presented a nomenclature that distinguishes causal relations as opposing or supporting and classifies them on a weak to strong spectrum. The rationales for assigning a strength value to a particular relation can be summarised such:

Strong - *causes & prevents*

The affector directly, or through automatic events, impacts the preconditions to an automatic event.

Weak - *enables & excludes*

The affector directly, or through automatic events, impacts the preconditions to a volition.

Weaker - *helps to enable & helps to exclude*

The affector impacts the preconditions to an event through volitions.

Weakest - *impedes preventing or excluding & impedes enabling*

The affector impacts an event that could impact the preconditions to another event.

The resulting full account of causal transitivity is then seen in table 7.3, with supporting relations denoted by + and opposing ones by –. It should be noted that rules of causality and transitivity are agent-neutral, that is, there is no different treatment applied to cases in which I help to enable my own action or help to enable someone else's action. It is in the specification of ethical restrictions that such matters can be distinguished. We finally note that the rules of transitivity proposed here are somewhat tentative and open to modification. The aim has been to formulate a systematic language that enables us to reason about the variety of relations that can characterise a particular domain and set of simulations, with a will to capture within different concepts the diverging mechanisms at play and degrees of causal influence. We acknowledge that further or different rules might be created to respond to this challenge in a different way.

Table 7.3: Transitivity of causal relations

Chain	**Transitivity**	**Polarity**
causes(E1,E2),causes(E2,E3)	causes(E1,E3)	+
causes(E1,E2),enables(E2,E3)	enables(E1,E3)	+
causes(E1,E2),prevents(E2,E3)	prevents(E1,E3)	–
causes(E1,E2),excludes(E2,E3)	excludes(E1,E3)	–
enables(E1,E2),causes(E2,E3)	helps(enables(E1,E3))	+
enables(E1,E2),enables(E2,E3)	helps(enables(E1,E3))	+
enables(E1,E2),prevents(E2,E3)	helps(excludes(E1,E3))	–
enables(E1,E2),excludes(E2,E3)	helps(excludes(E1,E3))	–
prevents(E1,E2),could cause(E2,E3)	prevents(E1,E3)	–
prevents(E1,E2),could enable(E2,E3)	excludes(E1,E3)	–
prevents(E1,E2),could prevent(E2,E3)	impedes(prevents(E1,E3))	+
prevents(E1,E2),could exclude(E2,E3)	impedes(excludes(E1,E3))	+
excludes(E1,E2),could cause(E2,E3)	impedes(enables(E1,E3))	–
excludes(E1,E2),could enable(E2,E3)	impedes(enables(E1,E3))	–
excludes(E1,E2),could prevent(E2,E3)	impedes(excludes(E1,E3))	+
excludes(E1,E2),could exclude(E2,E3)	impedes(excludes(E1,E3))	+

7.2.6 Causal Trace

To conclude the section on *event-based causality*, we present a formal definition of a 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 18 (Causal trace)**.** Given a *planning context* Ctx and a *causal motor* $\mathbb{C}_i$, the *causal trace* of a scenario $\sigma = (s, P)$ is defined as

$$ctr_{Ctx,i}(\sigma) = Th_{\mathcal{L}_r}(\mathbb{A}_{Ctx} \cup \mathbb{C}_i \cup P)$$

where $\mathcal{L}_r$ =r(s,$\mathcal{R}_i$,$\mathcal{E}$,$\mathcal{T}$,$\mathcal{E}$) with $\mathcal{R}_i$ = {causes, enables, prevents, excludes, helps(enables), helps(excludes), impedes(prevents), impedes(excludes), impedes(enables)}.

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. Throughout the following sections we systematically represent the *affector* by ϕ and the *end-state* by ψ.

7.3 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 can be 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 counterfactual and conditional tests. The assisted suicide of a terminally ill patient is typically less reprehensible than other forms of homicide because, among other things, 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 counterfactual assessment. Empirically, counterfactual thinking has widely been shown to play an important role in moral reasoning (e.g. [68, 170, 231]), though, as discussed, this property cannot alone account for responsibility attribution.

Given an affector volition ϕ, we now turn to exploring three counterfactual 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. Because ethical responsibility pertains to agents' choices, we consider only actions and omissions, rather than automatic events, as potential affectors.

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. $\Sigma^{\sigma \to t, \phi}$ is the set of all scenarios of $\Sigma^{\sigma \to t}$ that contain performs(s,ϕ,t) and reciprocally $\Sigma^{\sigma \to t, \overline{\phi}}$ is the set of all scenarios of $\Sigma^{\sigma \to t}$ that do not contain performs(s,ϕ,t).

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. $\Sigma_{t+}(\overline{\varepsilon}) = \bigcap_{t_2 \in \mathcal{T}, t_2 \geq t} \Sigma(\overline{\varepsilon}, t_2)$).

Counterfactual validity Is ψ counterfactually 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 19 (Counterfactual validity)**.** Given $\sigma = (s_n, P) \in \Sigma(\phi, t)$ such that $\sigma \models \phi \longmapsto \psi$, $\phi \longmapsto \psi$ is counterfactually valid iff $ov(\sigma, \phi, t) \in \Sigma_{t+}(\overline{\psi})$, where $ov(\sigma, \phi, t) \in \Sigma^{\sigma \to t, \overline{\phi}}$ is a unique completion of σ that overturns ϕ while keeping all other performed actions of σ that remain possible (see 7.3.1 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 20 (Cruciality)**.** Given $\sigma = (s_n, P) \in \Sigma(\phi, t)$ 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 21 (Extrinsic necessity)**.** Given $\sigma = (s_n, P) \in \Sigma(\phi, t)$ 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 22 (Elicited necessity)**.** Given $\sigma = (s_n, P) \in \Sigma(\phi, t)$ such that $\sigma \models \phi \longmapsto \psi$, ϕ *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 counterfactual 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.

7.3.1 Simple Counterfactual Validity

(Case 1. Counterfactual validity) If the state had granted him asylum status, he would not have committed suicide.
(Case 2. ¬ Counterfactual validity) If the state had granted 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 counterfactually 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. Within our framework, the counterfactual test cannot be applied to all affector volitions, in particular, it cannot be applied to all omissions. An omission that occurs in a simulation at a time when more than one action was available to the agent cannot be overturned. The opposite of an action is without ambiguity an omission, but when multiple actions are available, the opposite of an omission is not tractable. This is so in real life; the overturning of an action *dropping a glass* is *omitting to drop the glass*. But the overturning of an omission can be any one of the actions were available to an agent at the time of an omission. This point is similar to the argument that omissions should not be considered to prevent what other available actions could have caused: the existence of multiple available actions means that conceptualising the opposite of an omission is necessarily ambiguous. As such, we will only apply the counterfactual test to all actions and to omissions that occur when only a single action could have been performed in its place.

Finally, we note that this simple counterfactual test is one in which we imagine the affector to be overturned, 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 counterfactual 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. Recalling Lewis's view that a counterfactual is valid only if the world contradicting the affector that is the most similar to the actual world also contradicts the end-state, we might understand our method as one which takes the simplifying stance of considering 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 counterfactual). Figure 7.4 gives a visual representation of counterfactual validity and invalidity.

Modelling Simple counterfactual Validity

We have $\phi \longmapsto \psi$ in s_n, and we seek to establish whether ψ is true in $ov(\sigma_n, \phi, t)$.

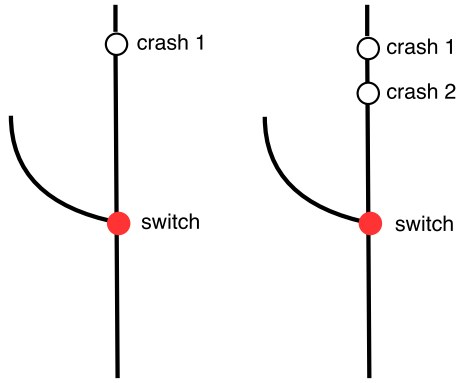

Figure 7.4: Counterfactual validity represented

Switch prevents crash 1. C-F Validity (L), ¬ C-F Validity (R).

In order to achieve this, we must first infer for every affector ϕ that occurs at some point in time in simulation s_n which is its corresponding simulation $ov(\sigma_n, \phi, t)$. As mentioned above, we here consider as affectors only actions and omissions that occur when only a single action could have been performed in its place. We also do not apply rules of scenario-based causality to automatic events because such events do not engage agent responsibility without some appeal to volitions.

Relative to actions, we state that for any action A occurring in a simulation S at a time T, the simulation in which A is overturned is the one

(a) which has the same history as S up to T,

(b) in which an omission O occurs at T (instead of A), and

(c) in which all actions that were performed in S after T are also performed if they are still complete after the occurrence of O, and in which no other actions are performed that were not performed in S.

Relative to omissions, we state that for any omission O occurring in a simulation S at a time T, a simulation in which O is overturned exists if there is no uncertainty in alternatives. Uncertainty exists at a T in S as soon as two actions are complete at T in S. We then state that the simulation in which O is overturned is the one

(a) which has the same history as S up to T,

(b) in which an action A occurs at T (instead of O), and

(c) in which all actions that were performed in S after T are also performed if they are still complete after the occurrence of A, and in which no other actions are performed that were not performed in S.

```
unequalFuture(T2,S1,S2):-
    occurs(S1,A,T1),complete(S2,A,T1),not occurs(S2,A,T1),action(A),time(T2),T2<T1.
unequalFuture(T2,S1,S2):-
    not occurs(S1,A,T1),occurs(S2,A,T1),sim(S1),action(A),time(T2),T2<T1.
equalFuture(T,S1,S2):-not unequalFuture(T,S1,S2),time(T),sim(S1;S2).
ovSim(S1,T,A1,S2):-
    occurs(S1,A1,T),occurs(S2,O,T),sameHistory(T,S1,S2),equalFuture(T,S1,S2),
    action(A1),omission(O).
uncertainty(S,T):-complete(S,A1,T),complete(S,A2,T),A1!=A2,action(A1;A2).
ovSim(S1,T,O,S2):-
    occurs(S1,O,T),occurs(S2,A,T),sameHistory(T,S1,S2),equalFuture(T,S1,S2),
    action(A),omission(O),not uncertainty(S1,T).
```

Ensuring that an event-based causal relation already holds between ϕ and ψ means we only submit to this test pairs of events that are already shown to be somewhat causally related. The `allSupRel(R)` and `allOppRel(R)` predicates distinguish supporting and opposing causal relations, be they direct or transitive.

Relative to supporting causality, we state that E is not counterfactually dependent on I if:

(a) I causes, enables, helps to enable, impedes preventing or impedes excluding E in S,

(b) E can arise in the simulation where I is overturned.

Relative to opposing causality, we state that E is not counterfactually dependent on I if:

(a) I prevents, excludes, helps to exclude or impedes enabling E in S,

(b) E never arises in the simulation where I is overturned.

From these rules, we derive the relations that are counterfactually valid.

```
notValid(S1,R,I,T,E):-
    r(S1,R,I,T,E),ovSim(S1,T,I,S2),canArise(S2,E),volition(I),event(E),allSupRel(R).
notValid(S1,R,I,T,E):-
    r(S1,R,I,T,E),ovSim(S1,T,I,S2),not canArise(S2,E),volition(I),event(E),allOppRel(R).
valid(S1,R,I,T,E):-
    r(S1,R,I,T,E),ovSim(S1,T,I,S2),not notValid(S1,R,I,T,E),volition(I),event(E).
```

Preemption and Over-determination

Because of *event-based* definitions of causality, the framework as it is can trace relations of causality even if they are counterfactually invalid. It will identify the relation through a `r(S,R,E1,T,E2)` predicate, then determine that it is counterfactually invalid. But some cases of invalidity, which

typically suggests lesser moral responsibility for the affector agent upon the outcome, might hide a set of dynamics that in fact suggest maintained moral responsibility. Instances of preemption and over-determination are such cases. Within these, we can also make a further distinction between the preemption and over-determination of an outcome *by other agents* - as in the bottle shattering example mentioned above - and *by the same agent*. If the presence of preemption or over-determination by other agents already fails to challenge moral responsibility (as in the fact that Suzy is at least to some extent responsible for the bottle shattering even if it was preempted by Billy), their presence resulting from a single agent does so even more firmly. Consider a burglar who is firmly set on raiding a house. He will try to break in through the back door, or, if this fails, through the front door which is sure to open but more exposed. If he breaks in through the back door, the fact that the house will be broken-into will not be counterfactually dependent on this action. Yet, clearly, the burglar if fully to blame. He simply preempts his own action. To handle such cases, we therefore postulate a number of further rules.

We state that an end-state is *over-determined by the same agent* G in S if

(a) G performs two actions which have a causal relation of the same polarity (opposing or supporting) to an end-state,

(b) these two actions are not counterfactually dependent on one another.

We state that an end-state is *over-determined by two agents* G1 and G2 in S if

(a) G1 and G2 each perform an action which has a causal relation of the same polarity (opposing or supporting) to an end-state,

(b) these two actions are not counterfactually dependent on one another.

Only actions are considered here as relevant affectors because the fact that an omission might lead the same end-state as an action simply means that the state of the world itself leads to the same end-state as the action, which corresponds to a genuine case of counterfactual invalidity. Next, we allow that the causal relations that link each action to the same end-state be different as long as they are both either supporting or opposing. Typical examples of over-determination pertain to identical relations; for instance if Suzy and Billy throw rocks at the same time, they both *cause* the bottle shattering. But this need not be true in all relevant cases. Take the bank robbery example given in section 7.2.5: I lend my car to my friend so he can drive the bank robber to the bank, which means I *help to enable* the heist. If, now, another robber had decided to rob the same bank on the same day, he would have *caused* the heist and over-determined my relation to it at the same time. Finally, we ensure that the two actions are not counterfactually dependent on one another so that we do not consider them to be separate independent causes when in fact one relies on the other to occur, as is typically the case within plans of action. We do not, for instance, want to consider that my lending of the car and the fact that my friend drove the car over-determine one

another, even though they both have a supporting relation to the heist: they are just part of the same causal chain. The corresponding rules are as follows.

```
samePolarity(R1,R2):-allOppRel(R1;R2).
samePolarity(R1,R2):-allSupRel(R1;R2).
sameAgent(act(G,X1),act(G,X2)):-occurs(S1,act(G,X1),T1),occurs(S2,act(G,X2),T2).
overdeterminedBySelf(S,R1,A1,A2,T1,E):-
    r(S,R1,A1,T1,E),r(S,R2,A2,T2,E),{valid(S,R3,A1,T1,A2):allSupRel(R3)}0,
    {valid(S,R3,A2,T2,A1):allSupRel(R3)}0,samePolarity(R1,R2),sameAgent(A1,A2),
    action(A1;A2),A1!=A2.
overdeterminedByOthers(S,R1,A1,A2,T1,E):-
    r(S,R1,A1,T1,E),r(S,R2,A2,T2,E),{valid(S,R3,A1,T1,A2):allSupRel(R3)}0,
    {valid(S,R3,A2,T2,A1):allSupRel(R3)}0,samePolarity(R1,R2),not sameAgent(A1,A2),
    action(A1;A2),A1!=A2.
```

Next, we state that an end-state is *preempted by a single agent* G in S1 if

(a) G performs an action A1 in S1 which has a causal relation to an end-state,

(b) G performs an action A2 in S2 which has a causal relation of the same polarity to the same end-state,

(c) S2 is the simulation in which A1 has been overturned.

We state that an end-state is *preempted by two agents* G1 and G2 in S1 if

(a) G1 performs an action A1 in S1 which has a causal relation to an end-state,

(b) G2 performs an action A2 in S2 which has a causal relation of the same polarity to the same end-state,

(c) S2 is the simulation in which A1 has been overturned.

```
preemptedBySelf(S1,R1,A1,A2,T1,E):-
    r(S1,R1,A1,T1,E),r(S2,R2,A2,T2,E),ovSim(S1,T1,A1,S2),samePolarity(R1,R2),
    sameAgent(A1,A2),action(A1;A2),T1<=T2.
preemptedByOthers(S1,R1,A1,A2,S2,T1,E):-
    r(S1,R1,A1,T1,E),r(S2,R2,A2,T2,E),ovSim(S1,T1,A1,S2),samePolarity(R1,R2),
    not sameAgent(A1,A2),action(A1;A2),T1<=T2.
```

Because of the way in which we define `ovSim(S1,T,I,S2)`, the framework in its present state will never find cases of preemption. Indeed, the simulation in which a volition is overturned is one in which no extra actions occur compared to the original simulation. So far we've investigated the overturning of volitions *all-things-equal*. In order to allow for cases of preemption and modelling propositions of the form "if agent G1 does not perform A1 and all else is equal, then agent G2 will

necessarily perform A2," we must add some rules which (a) force the performance of volitions by agents under certain circumstances and (b) update the definition of `ovSim(S1,T,I,S2)` to allow for additional actions. These may, for instance, take the following form.

```
% identifies actions that will be replaced, here, suzy's throw
replaces(act(suzy,throw),act(billy,throw)).
% forces either billy or suzy to throw the bottle
:-not occurs(S,act(suzy;billy,throw),T),complete(S,act(suzy;billy,throw),T).
% updates the definition of unequalFuture to not take into account cases of replacement
unequalFuture(T2,S1,S2):-
    not occurs(S1,A1,T1),occurs(S2,A1,T1),sim(S1),action(A1;A2),time(T2),T2<T1,
    {occurs(S1,A1,T1),replaces(A2,A1)}0.
% adds a new definition of ovSim to allow cases in which another action occurs in the
% ovSim instead of an omission
ovSim(S1,T,A1,S2):-
    occurs(S1,A1,T),occurs(S2,A2,T),sameHistory(T,S1,S2),equalFuture(T,S1,S2),
    replaces(A1,A2),action(A1;A2).
```

These new rules then give us the liberty to subtract from cases of counterfactual invalidity instances of preemption and over-determination by simply adding clauses of the kind "`not preemptedBySelf(S,R,A1,A2,T,E)`" on the right side of the definition of counterfactual invalidity.

7.3.2 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 individually sufficient 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 his 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 behaviour, for instance by pushing him to act more carefully. Figure 7.5 gives a visual representation of crucial and non-crucial causal relations.

One related but separate issue concerns "at-a-time" cases, where multiple agents or affectors participate together and at the same time towards a particular end-state [176]. This is true for such things as air pollution or noise from chatter in a classroom. These cases raise a number of questions, such as whether it is morally worse to be a contributor to a harm among a few others or among hundreds of others. Does it also matter whether every agent's contribution is necessary to produce the harm such that it is only as a group that their contribution is jointly sufficient? Or whether each agent's contribution is in itself sufficient? Or whether individual contributions are neither necessary nor sufficient to reach a certain relevant threshold of harm done? These cases are different from questions of cruciality in that cruciality looks at the role of potential affectors in turn, whereas "at-a-time" cases look at the role of actual affectors within a wider set of affectors. We do not model these here, but they make for compelling avenues of future work.

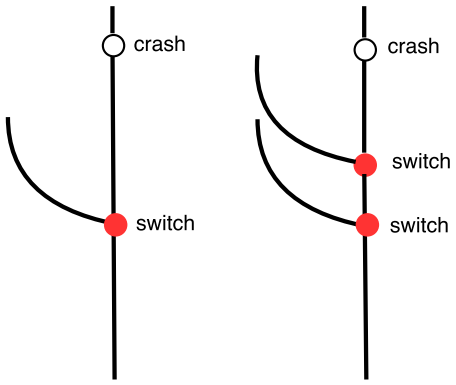

Figure 7.5: Cruciality represented

Switch prevents crash. Cruciality (L), ¬ Cruciality (R).

Modelling Cruciality

We have $\phi \longmapsto \psi$ in s_n and we seek to establish whether there exists a simulation in which ψ is true but ϕ isn't.

It is essential to note here 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,S1,I,S2)`, which selects for each I occurring at T in s_1 all the simulations corresponding to scenarios in $\Sigma^{\sigma_0 \to T,\overline{I}}$, i.e. simulations in which I does not occur and that are, up to T, identical to s_1 ('exc' standing for '*exclusive* of I').

```
laterSim(exc,S1,I,S2):-
    occurs(S1,I,T),not occurs(S2,I,T),sameHistory(T,S1,S2),volition(I),sim(S2).
```

Relative to supporting causality, we then state that I is not crucial to E if:

(a) I causes, enables, helps to enable, impedes preventing or impedes excluding E in S,

(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, excludes, helps to exclude or impedes enabling E in S,

(b) there is a later simulation exclusive of I in which E cannot arise.

From these rules, we infer relations of cruciality.

```
notCrucial(S1,R,I,T,E):-
    r(S1,R,I,T,E),laterSim(exc,S1,I,S2),canArise(S2,E),volition(I),event(E),allSupRel(R).
notCrucial(S1,R,I,T,E):-
    r(S1,R,I,T,E),laterSim(exc,S1,I,S2),not canArise(S2,E),volition(I),event(E),
    allOppRel(R).
crucial(S1,R,I,T,E):-
    r(S1,R,I,T,E),not notCrucial(S1,R,I,T,E),volition(I),event(E).
```

7.3.3 Extrinsic Necessity

> *(Case 5. Extrinsic necessity)* If I hadn't picked the money up from the floor, someone else would have, and the owner would never had been able to retrieve it.
>
> *(Case 6. ¬ Extrinsic necessity)* If I hadn't picked the money up from the floor, no one else would have, and the owner might have been able to retrieve it.

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 extrinsic 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, though we may additionally want the indictment of the mob boss. Whatever the situation, it is often important to investigate such circumstantial aspects. Figure 7.6 gives a visual representation of extrinsically necessary and non-extrinsically necessary causal relations.

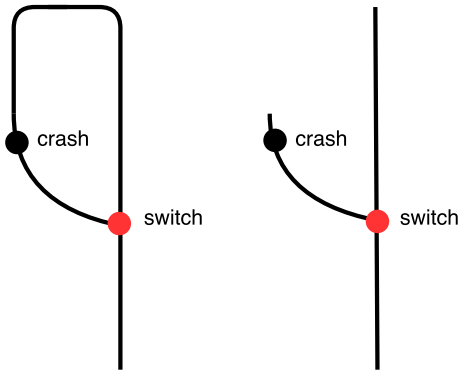

Figure 7.6: Extrinsic necessity represented

Switch causes crash. Extrinsic Necessity (L), ¬ Extrinsic Necessity (R).

Modelling Extrinsic Necessity

We have $\phi \longmapsto \psi$ in s_n 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 would have necessarily arisen after that time had the affector volition not occured, regardless of what every agent does. To model it, we employ the `laterSim(exc,S1,I,S2)` predicate defined above.

Relative to supporting causality, we then state that E is not extrinsically necessary relative to I if:

(a) I causes, enables, helps to enable, impedes preventing or impedes excluding E in S,

(b) there exists a later simulation exclusive of I in which E cannot arise.

Relative to opposing causality, we state that E is not extrinsically necessary relative to I if:

(a) I prevents, excludes, helps to exclude or impedes enabling E in S,

(b) there exists a later simulation exclusive of I in which E can arise.

From these rules, we infer relations of extrinsic necessity.

```
notNecessary(S1,R,exc,I,T,E):-
    r(S1,R,I,T,E),laterSim(exc,S1,I,S2),not canArise(S2,E),volition(I),event(E),
    allSupRel(R).
notNecessary(S1,R,exc,I,T,E):-
    r(S1,R,I,T,E),laterSim(exc,S1,I,S2),canArise(S2,E),volition(I),event(E),allOppRel(R).
necessary(S1,R,exc,I,T,E):-
    r(S1,R,I,T,E),not notNecessary(S1,R,exc,I,T,E),volition(I),event(E).
```

7.3.4 Elicited Necessity

(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. Figure 7.7 gives a visual representation of causal relations in which the end-state is necessarily elicited and in which it is not necessarily elicited.

Modelling Elicited Necessity

We have $\phi \longmapsto \psi$ in s_n 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,S1,I,S2)`, which selects for each I occurring at T in s_n all the simulations corresponding to scenarios in $\Sigma^{\sigma_n \to T,I}$, i.e. simulations in which I occurs and that are, up to T, identical to s_n.

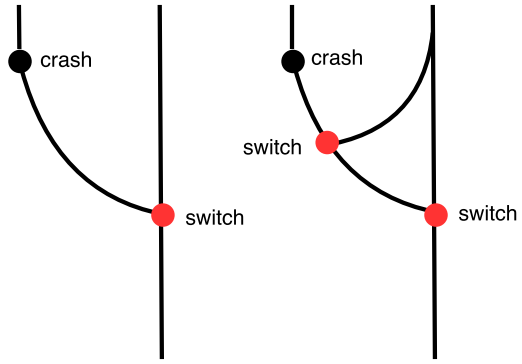

Figure 7.7: Elicited necessity represented

Switch causes crash. Elicited Necessity (L), ¬ Elicited Necessity (R).

```
laterSim(inc,S1,I,S2):-occurs(S1,I,T),occurs(S2,I,T),sameHistory(T,S1,S2),volition(I).
```

Relative to supporting causality, we then state that I does not make E necessary if:

(a) I causes, enables, helps to enable, impedes preventing or impedes excluding E in S,

(b) there exists a later simulation inclusive of I in which E cannot arise.

Relative to opposing causality, we state that I does not make E necessary if:

(a) I prevents, excludes, helps to exclude or impedes enabling E in S,

(b) there exists a later simulation inclusive of I in which E can arise.

From these rules, we infer relations of elicited necessity.

```
notNecessary(S1,R,inc,I,T,E):-
    r(S1,R,I,T,E),laterSim(inc,S1,I,S2),not canArise(S2,E),volition(I),event(E),
    allSupRel(R).
notNecessary(S1,R,inc,I,T,E):-
    r(S1,R,I,T,E),laterSim(inc,S1,I,S2),canArise(S2,E),volition(I),event(E)
    ,allOppRel(R).
necessary(S1,R,inc,I,T,E):-
    r(S1,R,I,T,E),not notNecessary(S1,R,inc,I,T,E),volition(I),event(E).
```

Correspondence between Extrinsic and Elicited Necessity

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 one's volition. These two properties can be combined to generate four cases with sharply distinct ramifications:

¬ extrinsic necessity & ¬ elicited necessity This is a neutral case relative to these properties in that the end-state is avoidable both before and after the considered affector volition occurs.

extrinsic necessity & elicited necessity This 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). Whatever any agent does from the time of the considered volition, the considered end-state will eventually come true. Here, affector responsibility over the end-state is diminished if not nullified.

¬ extrinsic necessity & elicited necessity This 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.

extrinsic necessity & ¬ elicited necessity Finally, this case 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 the considered simulation, 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.

This correspondence between the two properties also means that they can be modelled together, as long as the conditions of exclusivity and inclusivity are identified separately. We can do this by simply replacing the 'exc' and 'inc' conditions by a variable and employ a single set of rules for both types of necessity.

7.3.5 Dependencies

The counterfactual properties defined above are partially interdependent, and in part mutually exclusive. For example, if ϕ causes ψ in a way that was extrinsically necessary, then inevitably the relation is counterfactually 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 7.4 with properties organised in descending order of stringency, from cruciality to simple counterfactual validity to extrinsic necessity (E.N.). The instances a, b, c and d represent the four possible combinations of counterfactual properties, impossible ones being greyed out. Elicited necessity is not comparable to the others in these terms because it is a property that is not counterfactual, as it investigates the simulations in which the considered affector does occur. Another similar aspect we can note is that some configurations within a scenario can make

certain properties automatically true. For example, any time an affector produces an end-state without there being any possible volitions between the two, the relationship will automatically be one of elicited necessity.

Table 7.4: Dependencies between counterfactual properties

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	

In addition, we note that some of the relations defined here supply different ways of appraising the same dynamic. In particular, the fact that an action prevents an event in a way that is counterfactually valid corresponds to the fact that an omission occurring instead (at the same time in a simulation with the same history) *causes* that event.

Pertinence of the Same History Problem for Scenario-Based Causality

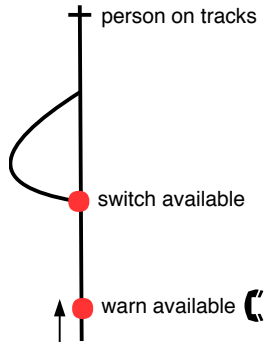

Figure 7.8: *Same history problem*: Domain

Here we take a look at the way in which the same history problem raises issues for the above properties. It manifests itself because in order to investigate them, the agent needs to compare a particular plan with other relevant plans in the domain. Consider the following. A train is running towards a bystander standing on the tracks. The agent has two available actions, warning the bystander at t=1 and pressing a switch at t=2. If the agent *warns*, then the bystander will leave the tracks, while the *switch* action only has the effect of temporarily driving away the train, because the side tracks loop back onto the main tracks. This results in four possible configurations in which 1, 2 or no action is performed. The domain and set of all possible scenarios are represented in figures 7.8 and 7.9.

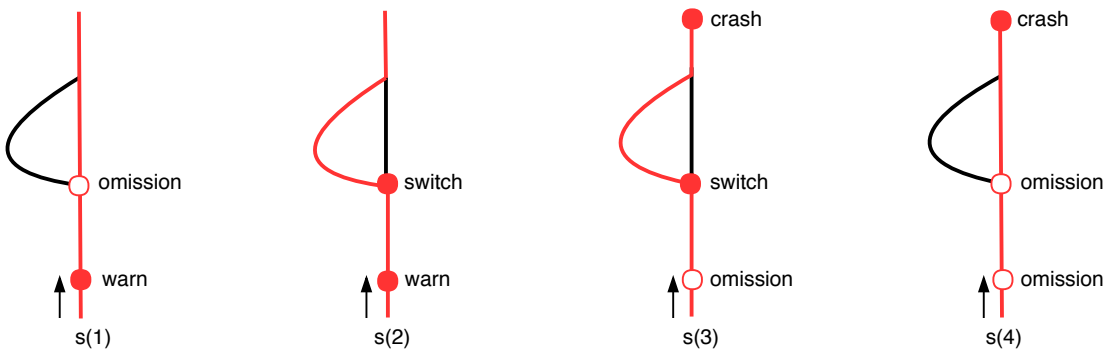

Figure 7.9: *Same history problem*: Set of simulations

The `sameHistory` predicate is necessary here because it separates into two distinct groups s(1)/s(2) and s(3)/s(4). This needs to be done because once *warn* has been performed, the crash cannot happen. If *warn* is not performed, however, the crash is obligatory. This means that in s(3) and s(4), performing the switch action or omitting to perform the switch action will invariably cause the crash in a way that is counterfactually invalid, extrinsically necessary and necessarily elicited. Without the presence of the `sameHistory` predicate within the definitions of `laterSim`, these relations would be seen to be potentially counterfactually valid and not necessary. The other two simulations would have been taken into account in the analysis, generating false results by considering that the crash could have been avoided even after no warning had been performed.

7.4 Discussion and Related Works

The purpose of the work presented in this chapter is twofold. First, all the distinctions made can be used to fine tune consequentialist theories of ethics, by helping to determine what indeed are the relevant causes and consequences of agents' volitions. Second, though it does not prescribe specific definitions of causal and moral responsibility, it provides a framework for modelling the properties that might make up such definitions, as proposed by logicians and philosophers. Here we give an example of how this might be done.

We can appeal to the properties of necessity to model the definitions of causal responsibility given by Lorini *et al.* in [157]. Using STIT logic, they define two kinds of causal responsibility. Informally, an agent is said to be *actively causally responsible* for an event if he sees to it that it is true, and *passively causally responsible* for an event if he could have prevented it from being true.

> [*Active causal responsibility*] is expressed by the so-called 'deliberative' STIT operator [i **dstit**] which is definable in the following way:

$$[i\ \mathbf{dstit}]\varphi \stackrel{def}{=} [i\ \mathbf{stit}]\varphi \wedge \langle \varnothing \mathbf{stit} \rangle \neg \varphi$$

> The construction $[i\ \mathbf{dstit}]\varphi$ can be read 'agent i is causally responsible for bringing about φ'. The negative condition $\langle \varnothing \mathbf{stit} \rangle \neg \varphi$ is given to prevent an agent from being causally responsible for φ, when φ is something inevitable (in the sense that it is true regardless of what every agent does).

We can look in turn at the various elements of this definition.

φ In constructions of the form $[i\ \mathbf{dstit}]\varphi$, the authors state that φ corresponds to a formula or a state of affairs. It may therefore be equated to a number of entities, including a single event occurrence, a single event non-occurrence, a combination of these, a state of the world, etc. Looking at this through the prism of our own framework, we reduce the meaning of φ to a *single* event instance, for that is what we have so far considered to be an *outcome* (the framework would need to be enhanced to handle outcomes of greater complexity). Assuming this, there are still two ways in which φ can be understood: it may be equated to an event ϵ, such that $[i\ \mathbf{dstit}]\varphi$ means *i sees to it that* ϵ *occurs*, or it may be read as a formula of either the form "event ϵ occurs" or "event ϵ does not occur," such that $[i\ \mathbf{dstit}]\varphi$ can either mean *i sees to it that* ϵ *occurs* or *i sees to it that* ϵ *does not occur.* Based on the definitions given in [119], we take the second approach, meaning that we will consider both supporting and opposing relations as possible bearers of active causal responsibility. For instance, *seeing to it that* φ, with φ equal to "ϵ does not occur" amounts to, among other things, preventing ϵ in our sense of the term. The terms "active" and "passive" do not here map onto the concepts of causing and preventing. Moreover, we consider that ϵ is an automatic event but not a volition, because there is no specific discussion of volitions as end-states in [157].

[*i* dstit] We translate the proposition "agent i sees to it that φ" as *agent i performs an action which has a causal relation to* φ *in a way that is intrinsically necessary.* The property of intrinsic necessity captures the essence of the deliberative **stit** operator from which it follows that "if H is a singleton i, the construction [i stit]φ has to be read 'agent i sees to it that φ no matter what the other agents do"'[157]. In other words, an agent sees to it that φ by acting at a point in time in such a way that the truth of φ is guaranteed [119].[1] Any type of relation might be considered, even the weaker kind, which in some cases display intrinsic necessity. Take the following example for

[1] We could adapt the definition of intrinsic necessity to make it correspond to another type of stit operator, the achievement stit (as described in [119]) by tracking the time of the end-state in addition to tracking the time of the affector action. For supporting relations, this would amount to requiring that a relation of *achievement* intrinsic necessity holds if the end-state that occurs in the original simulation occurs *at the same time* in all later simulations inclusive of the affector. For opposing relations, this would demand that we track the moment at which the avoided end-state could have occured, so that we may state that the property holds if the end-state is avoided in the original simulation at the same time as in all later simulations inclusive of the affector.

enabling: Kelly poisons her enemy Chad. This poison will bring a slow and painful death, which Chad knows, so he kills himself in a painless and quicker way instead. Here Kelly helps to enable the death by giving Chad a reason to kill himself, and this relation is intrinsically necessary because once the poison has been administered, Chad necessarily must die. Relative to affector volitions, the paper makes explicit reference to actions but does not discuss the possibility that an omission might be at the origin of causal responsibility. As such, we only consider actions as potential affectors. The idea that an agent sees to it that φ by omitting to act is arguably awkward, but it could be argued that as far as they also display agency, omissions should also be apprehended as potential affectors. Computationally, this would simply amount to replacing the A variables with I variables in the upcoming rules.

$\langle \varnothing \mathbf{stit} \rangle \neg \varphi$ We translate the requirement that the end-state must not be inevitable by simply stating that the relation must not be extrinsically necessary. We already postulate that it is intrinsically necessary, hence the adition of extrinsic necessity would yeild absolute necessity, i.e. inevitability. The resulting ASP rule for Lorini *et al.*'s *active causal responsibility* is the following.

```
activeCR(S,R,A,T,U):-
    necessary(S,R,inc,A,T,U),notNecessary(S,R,exc,A,T,U),action(A),auto(U).
```

Relative to passive causal responsibility, the following definition is given.

> As for *passive causal responsibility*, we say that an agent i is passively causal responsible for letting φ be true if and only if 'agent i could have prevented φ from being true', denoted by CHP(i,φ). In E-GSTIT logic the construction CHP(i,φ) can be decomposed by assuming that agent i could have prevented φ from being true if and only if: (1) φ is true in the actual world (i.e. φ) and (2) given what the others have chosen to do, if i had chosen differently then φ would have necessarily been false (i.e. $\langle$Agt$\backslash\{$it$\}$**stit**$\rangle$ [Agt **stit**]$\neg\varphi$)[2][...].

$$\text{CHP(i,}\varphi\text{)} \stackrel{def}{=} \varphi \wedge \langle \text{Agt} \backslash \{\text{it}\} \mathbf{stit} \rangle \ [\text{Agt } \mathbf{stit}] \neg \varphi$$

Because we consider that φ is a formula of the form "event ϵ occurs" or "event ϵ does not occur," the notion of preventing φ from being true can mean one of two things: preventing the occurrence of an event or preventing the fact an event does not occur. As such, given an automatic event ϵ, if φ is a proposition of the form "ϵ occurs", then agent i is passively causally responsible for φ at a

[2] As made clear in the remainder of the paper, "if i had chosen differently then φ would have necessarily been false" is to be read as "there exists an action available to i such that φ would have necessarily been false" rather than "whichever other action i chose then φ would have necessarily been false."

time T in S if:

(a) i chooses a volition V at T in S (this can be an action or an omission),

(b) ϵ occurs at a time later than T in S,

(c) there exists another simulation that has the same history as S up until T in which a different action A performed by i has an opposing and inclusively necessary relation with ϵ.

Given an automatic event ϵ, if φ is a proposition of the form "ϵ does not occur", then agent i is passively causally responsible for φ at a time T in S if:

(a) i chooses a volition V at T in S,

(b) ϵ does not occur in S,

(c) there exists another simulation that has the same history as S up until T in which a different action A performed by i has a supporting and inclusively necessary relation with ϵ. The resulting ASP rules for passive causal responsibility are the following.
In both cases, we must ensure that it is the same agent who does the initial volition and the potential action which would prevent φ.

```
sameAgent(omit(G,X1),act(G,X2)):-occurs(S1,omit(G,X1),T1),occurs(S2,act(G,X2),T2).
passiveCR(S1,occurence,I,T1,U):-
    occurs(S1,I,T1),occurs(S1,U,T2),T1<T2,necessary(S2,R,inc,A,T1,U),allOppRel(R),
    sameHistory(T1,S1,S2),sameAgent(I,A),auto(U),volition(I),action(A),I!=A.
passiveCR(S1,nonOccurrence,I,T1,U):-
    occurs(S1,I,T1),not occurs(S1,U,T2),time(T2),necessary(S2,R,inc,I2,A,U),allSupRel(R),
    sameHistory(T1,S1,S2), sameAgent(I,A),auto(U),volition(I),action(A),I!=A.
```

As a final note, if we were to consider that φ was an event rather than a proposition, the corresponding ASP rules would be:

```
activeCRE(S,A,T,U):-
    necessary(S,R,inc,A,T,U),notNecessary(S,R,exc,A,T,U),AllSupRel(R),action(A),auto(U).
passiveCRE(S1,I,T1,U):-
    occurs(S1,I,T1),occurs(S1,U,T2),T1<T2,necessary(S2,R,inc,A,T1,U),AllOppRel(R),
    sameHistory(T1,S1,S2),sameAgent(I,A),auto(U),volition(I),action(A),I!=A.
```

7.5 Proof of Concept #2 (*collision*)

We now demonstrate the use of the causal concepts described here through a realistic toy example, appealing to the (*collision*) example described in section 6.3. We unpack simulation s_4. As we

recall, this is the simulation in which Iris brings the driver to safety and calls for help. The driver then partly recovers, but fails to go to rehab and ends up with a limp. The *event trace* of the simulation is the following (keeping in mind that for all fluents which do not hold, their negation does).

```
sim(s(4)).
% t=0
holds(s(4),breathing(driver),0).
holds(s(4),hurt(driver),0).
occurs(s(4),act(iris,safety(driver)),0).
% t=1
holds(s(4),breathing(driver),1).
holds(s(4),hurt(driver),1).
holds(s(4),safe(driver),1).
occurs(s(4),act(iris,aid(driver)),1).
% t=2
holds(s(4),breathing(driver),2).
holds(s(4),safe(driver),2).
holds(s(4),treatment(driver),2).
occurs(s(4),omit(iris,2),2).
% t=3
holds(s(4),breathing(driver),3).
holds(s(4),safe(driver),3).
holds(s(4),treatment(driver),3).
occurs(s(4),partRecovery(driver),3).
% t=4
holds(s(4),breathing(driver),4).
holds(s(4),safe(driver),4).
holds(s(4),needRehab(driver),4).
occurs(s(4),omit(driver,4),4).
% t=5
holds(s(4),breathing(driver),5).
holds(s(4),safe(driver),5).
holds(s(4),needRehab(driver),5).
occurs(s(4),limp(driver),5).
```

The *causal trace* is the following.

```
% t=0
r(s(4),enables,act(iris,safety(driver)),0,act(iris,aid(driver))).
r(s(4),prevents,act(iris,safety(driver)),0,accident(driver)).
r(s(4),prevents,act(iris,safety(driver)),0,death(driver)).
r(s(4),excludes,act(iris,safety(driver)),0,act(iris,kill(driver))).
r(s(4),excludes,act(iris,safety(driver)),0,omit(iris,1)).
% transitive relations
r(s(4),helps(enables),act(iris,safety(driver)),0,omit(iris,3)).
r(s(4),helps(enables),act(iris,safety(driver)),0,omit(driver,5)).
r(s(4),helps(enables),act(iris,safety(driver)),0,partRecovery(driver)).
r(s(4),helps(enables),act(iris,safety(driver)),0,limp(driver)).
r(s(4),helps(excludes),act(iris,safety(driver)),0,act(driver,rehab)).
r(s(4),helps(excludes),act(iris,safety(driver)),0,act(iris,call)).
r(s(4),helps(excludes),act(iris,safety(driver)),0,omit(iris,2)).
r(s(4),helps(excludes),act(iris,safety(driver)),0,decline(driver)).
r(s(4),impedes(enables),act(iris,safety(driver)),0,fullRecovery(driver)).
% t=1
r(s(4),causes,act(iris,aid(driver)),1,partRecovery(driver)).
r(s(4),causes,act(iris,aid(driver)),1,limp(driver)).
r(s(4),enables,act(iris,aid(driver)),1,omit(iris,3)).
r(s(4),enables,act(iris,aid(driver)),1,omit(driver,5)).
r(s(4),prevents,act(iris,aid(driver)),1,decline(driver)).
r(s(4),prevents,act(iris,aid(driver)),1,death(driver)).
r(s(4),excludes,act(iris,aid(driver)),1,omit(iris,2)).
% transitive relations
r(s(4),helps(excludes),act(iris,aid(driver)),1,act(iris,call)).
r(s(4),helps(excludes),act(iris,aid(driver)),1,act(driver,rehab)).
r(s(4),impedes(enables),act(iris,aid(driver)),1,fullRecovery(driver)).
% t=2
r(s(4),causes,omit(iris,2),2,partRecovery(driver)).
r(s(4),causes,omit(iris,2),2,limp(driver)).
r(s(4),enables,omit(iris,2),2,omit(driver,5)).
r(s(4),excludes,omit(iris,2),2,act(iris,call)).
% transitive relations
r(s(4),helps(excludes),omit(iris,2),2,act(driver,rehab)).
r(s(4),impedes(enables),omit(iris,2),2,fullRecovery(driver)).
% t=3
r(s(4),causes,partRecovery(driver),3,limp(driver)).
```

```
r(s(4),enables,partRecovery(driver),3,omit(driver,5)).
% transitive relations
r(s(4),helps(excludes),partRecovery(driver),3,act(driver,rehab)).
r(s(4),impedes(enables),partRecovery(driver),3,fullRecovery(driver)).
% t=4
r(s(4),causes,omit(driver,4),4,limp(driver)).
r(s(4),excludes,omit(driver,4),4,act(driver,rehab)).
% transitive relations
r(s(4),impedes(enables),omit(driver,4),4,fullRecovery(driver)).
```

We give a visual representation of this causal trace that is, for reasons of readability, spread across two figures. Non-transitive relations are represented in figure 7.10, relations of transitivity are represented in figure 7.11.

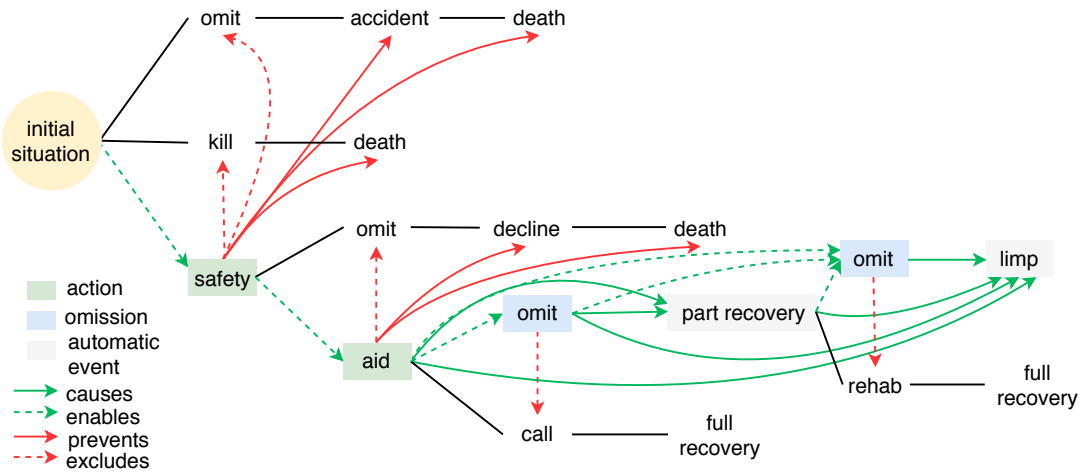

Figure 7.10: Proof of concept: Event-based causal relations (*collision*)

In order to illustrate in their variety the properties of scenario-based causality, we zoom out and take a look at all the simulations at once. In figure 7.12 we represent some interesting relations within the domain that are found in the answer set. For clarity, among the numerous properties of scenario-based causality inferred from the set of simulations, we focus on a small set of examples concerning caused outcomes.

counterfactual validity. If Iris decides to kill the driver when she pulls up to him, her action can arguably be considered poorly impacting since her causal relation to the death is *counterfactually invalid*. If she doesn't kill him and all else is kept equal (i.e. she performs no other

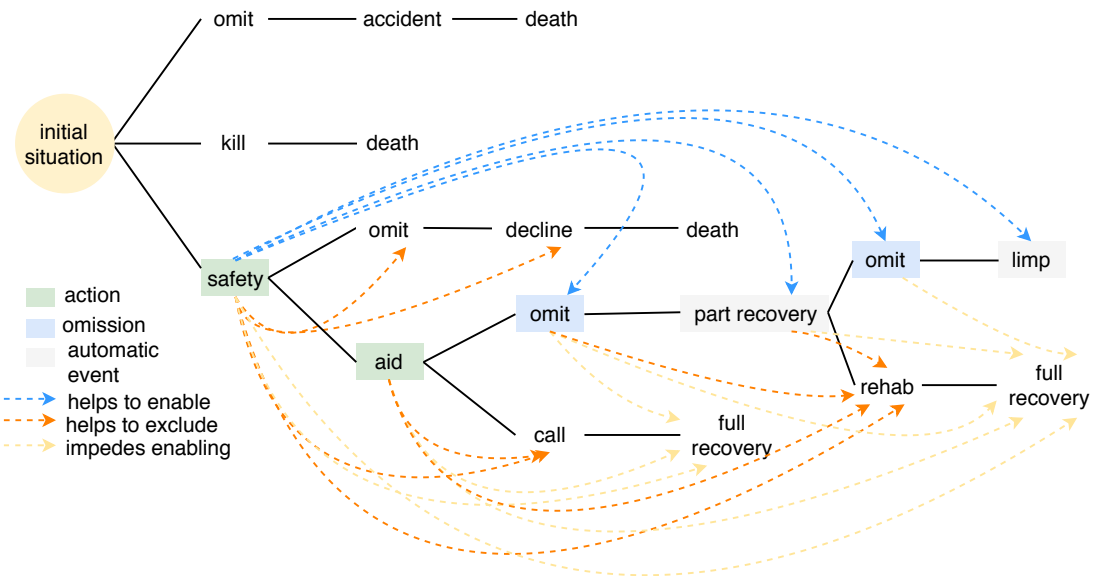

Figure 7.11: Proof of concept: Transitive event-based causal relations (*collision*)

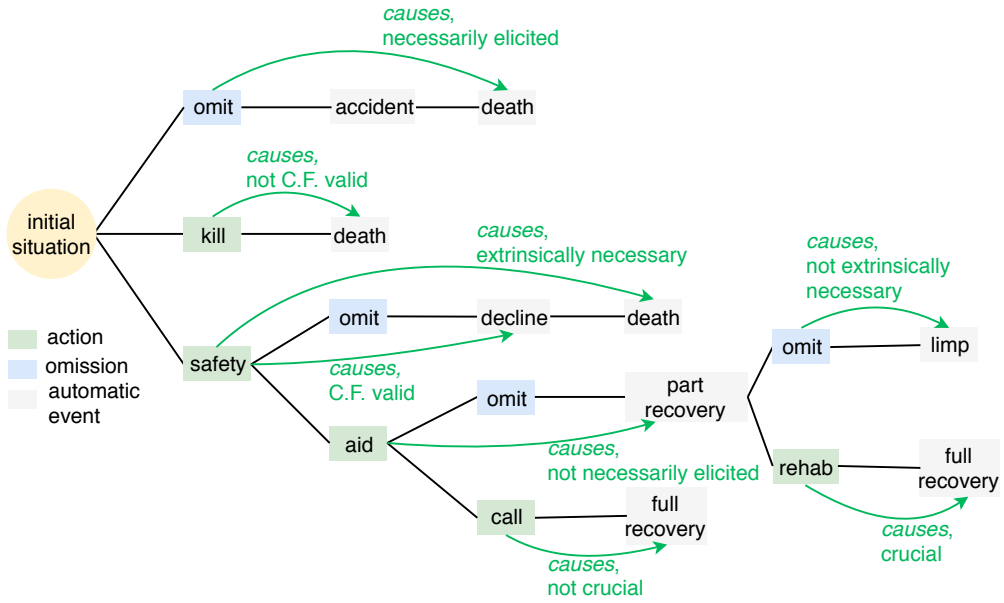

Figure 7.12: Proof of concept: Properties of scenario-based causality (*collision*)

actions), then the driver left on the road will be killed by another passing car. It is also true that bringing the driver to safety and doing nothing else causes the driver to die in a

counterfactually invalid way. However, the fact that in this case the driver dies because his health declines is *counterfactually valid*. Indeed, if Iris does nothing at all, the driver dies because of a new accident and not because of that decline. We see that if for some reason Iris could not perform first aid or call for help, the only aspects of the situation upon which she could have any influence would be the way in which the driver dies, but not the death itself.

Cruciality. If Iris brings the driver to safety, performs first aid and calls for help, the call for help will cause the driver to reach full recovery by ensuring hospital care. Calling for help, however, is *not crucial* to full recovery, since if Iris does not call, the driver can still reach full recovery if he goes to rehab. If Iris performs the first two actions but then fails to call for help and quickly get him to a hospital, however, going to rehab is *crucial* if the driver wants to fully recover after having partly done so.

Extrinsic necessity. Bringing the driver to safety without taking any further action causes the driver's death. This is true in a way that is *extrinsically necessary*, meaning that if Iris had performed any possible action other than bring the driver to safety, he would have died anyway (however, the relation is not one of elicited necessity, because bringing the driver to safety makes his salute possible). A *non-extrinsically necessary* relation of causality is found in the fact that if the driver partly recovers then performs no action, he will end up with a limp. This omission causes the limp, but, had he not omitted to act, he could have avoided it (by going to rehab).

Elicited necessity. Once Iris brings the driver to safety and performs first aid, she can cease her actions, leading the driver to partly recover. But her aiding action will cause this in a way that is *not necessarily elicited*, because she would have had the possibility of calling for help and allowing the driver to directly fully recover without going through painful partial recovery. In other cases, her choices ineluctably lead to certain outcomes. For instance, when she sees the driver at first, if she chooses not to act then she *necessarily elicits* the driver's death.

The corresponding lines in the answer set are the following.

```
% counterfactual validity
valid(s(3),causes,act(iris,safety(driver)),0,decline(driver)).
notValid(s(2),causes,act(iris,kill(driver)),0,death(driver)).
% Cruciality
crucial(s(5),causes,act(driver,rehab),4,fullRecovery(driver)).
notCrucial(s(6),causes,act(iris,call),2,fullRecovery(driver)).
% Extrinsic necessity
necessary(s(3),causes,exc,act(iris,safety(driver)),0,death(driver)).
notNecessary(s(4),causes,exc,omit(driver,4),4,limp(driver)).
```

```
% Elicited necessity
necessary(s(1),causes,inc,omit(iris,0),0,death(driver)).
notNecessary(s(4),causes,inc,act(iris,aid(driver)),1,partRecovery(driver)).
```

Part V

Model of the Good

Chapter 8

Contribution: A Model of the Good

In order to implement fully fledged *consequentialist* theories of the right, which place additional ethical valuation on actions with determined effects, we must first have an account of which of these effects are desirable and which are not. One way of doing this is by simply stating that, for instance, a train crash is undesirable and that people staying alive is desirable. Within our semantics, however, it is fitting and more systematic to evaluate the desirability of volitions in terms of the effects they have on relevant ethical *modalities*, such as human rights or guiding concepts. In this section, we therefore present three modes for defining the Good, one based on rights, one based on hedonistic motivations and one based on moral values[1]. The first two modes define the Good relative to how the consequences of a volition impact defined modalities, while the third focuses on how the volition itself impacts modalities. We call the first type *impact-centered* theories of the Good, and the second type *agent-centered* theories of the Good. Within each mode, we propose a way to quantify the Good once it has been qualified relative to a certain modality. This gives events meaningful weights and allows theories of the Good to be integrated within theories of the Right. The ways of qualifying and quantifying the Good presented bellow are interchangeable, open to modification and can also be combined.

[1]Note that *theories of the Right* and *rights as modalities* are false friends. The Right denotes the ethically correct while the rights within a theory of the Good denote individual principles of freedom or entitlement.

8.1 Impact-Centered Theories

8.1.1 Theory of Rights

Nozick's so called "utilitarianism of rights" posits that rights not being violated is constitutive of the Good to be maximised [182]. A right may be defined as a "*justified claim that individuals and groups can make upon other individuals or upon society; to have a right is to be in a position to determine by one's choices, what others should do or need not do*" [21]. This definition captures well the fact that a right denotes both a state of affairs for the person concerned (the exercise of the right) and a constraint imposed upon others (the prohibition of violating the right). As such, we consider that a given event can either respect a right, infringe it, or fail to impact it. In order to model a corresponding theory of the Good, we propose the following taxonomy, by accounting for:

- The number of people involved in events. For example, an event that affects five people will have a five times greater count than an event which affects one person. This information is given by the `targetNumber(G,N)` predicate as N. Here, G might represent a single person or a group of people.
- The relative value of the people involved in the event. Because an event might affect different people differently, we can separate different sets of agents into different target groups and appraise them independently. For example, it may be more significant to save children than adults, harm healthy people rather than declining patients, or benefit the poor rather than the rich. The given value is measured by assigning to each affected group a numerical weight, expressed by the `targetWeight(G,N)` predicate as N. The target number and target weight values are then concatenated into a single value within the `target(G,N)` predicate where N is their product. Target weights and target numbers compose the *target specification* module of a theory of the Good.
- The importance of the right affected by the event. For example, respecting the right to life may be more important than respecting the right to property. This is measured by assigning to each right a numerical weight, expressed by the `modalityWeight(M,N)` predicate where M is the modality and N the given weight. The measurement scale is unfixed, and may also be defined by preference relations whereby the preference of a modality over another will mean the former has a greater weight than the latter. We here specifically consider the modality of rights, but all modalities can be given weights throughout theories of the Good. Modalities and modality weights compose the *modality specification* module.
- The extent to which a given event impacts a right. An event can affect a right in one of two ways, it can support or negate it, but it can also do so to varying degrees. For example,

stealing a television and stealing a car will infringe the right to property to varying degrees. We represent this in the form of a percentage value, expressed as N in a predicate which also identifies the considered event, the impacted agents and the considered right. If an event E supports a right, the predicate will take the form of `impact(E,G,right,N)`, and if it negates a right, the predicate will take the form of `impact(E,G,neg(right),N)`. Again, this can be widened to other modalities beyond rights. The set of `impact` predicates in a given domain compose the *ethical event specification* module relative to impact-centered theories of the Good.

We then define rules such that an event which impacts at least one person (i.e. agent or pseudo-agent) and negates a right is bad in relation to that right, and an event which impacts at least one person and supports a right is good in relation to that right. An event may as such be bad in relation to a right and good in relation to another. The extent to which an event is then considered good or bad is a function of the importance given to the considered modality, the importance given to the target agent or agents, their number, and the strength of impact given as a percentage. An event which neither supports nor violates a right is neutral relative to that right.

```
target(G,N1*N2):-targetNumber(G,N1),targetWeight(G,N2).
good(E,M,G,N1*N2*N3/100):-
    impact(E,G,M,N1),modalityWeight(M,N2),target(G,N3),right(M).
bad(E,M,G,N1*N2*N3/100):-
    impact(E,G,neg(M),N1),modalityWeight(M,N2),target(G,N3),right(M).
```

8.1.2 Hedonism

Hedonism, broadly, is the utilitarian view that the moral worth of an action derives from the volume of pleasure it produces minus the volume of pleasure it negates (i.e. pain it produces). Versions of hedonism differ in the way they characterise pleasure as a more or less complex entity, composed, or not, of different types of sub-categories of pleasure and pain, such a well-being, happiness, advantage, or sadness, stress, loss, etc. All types of hedonism nevertheless agree in making the arguably bold statement that pleasure and pain are the only things that ultimately matter and are intrinsically valuable. It sets itself apart from many other ethical theories which place morality outside of immediate, personal feelings. Typically, both physical and mental phenomena are considered relevant, and no distinction is made between one's own pleasure and the pleasure of others. Unlike the frequent use of the word "hedonist" to mean self-serving and egotistical, philosophical hedonism pertains to the population as a whole; what matters is overall net pleasure. Jeremy Bentham and his protégé John Stuart Mill are two famous proponents of hedonistic utilitarianism [26, 172].

While the first appraises pleasure through various forms that are ultimately roughly synonymous (happiness, benefit, etc.), the second considers that some sources of pleasure are more worthy than others. He distinguishes lower pleasures derived from the body and shared with animals (such as pleasure from eating or from sex), from higher pleasures of the mind that are unique to humans (such as pleasure gained from appreciating art or from reflection).

We here propose a simple account of hedonism by considering that an event is good in so far as it produces pleasure in agents, and bad in so far as it negates it. A single event might be both good and bad in this view. As with the impact-centered theory of the Good based on rights, we take into account target weight, target number and force of impact, though not as a percentage but as a determined value corresponding directly to the intensity of the pleasure or pain caused.

```
bad(E,pleasure,G,N1*N2):-impact(E,G,neg(pleasure),N1),target(G,N2).
good(E,pleasure,G,N1*N2):-impact(E,G,pleasure,N1),target(G,N2).
```

In impact-centered theories of the Good, we systematically consider that modalities are beneficial entities, such as rights to be respected or pleasure to be maximised. This allows us to work on a single scale for each entity rather than to appeal to two concepts that fundamentally correspond to the two ends of a single spectrum. For example, we consider that pain is simply the negation of pleasure. This also allows us to infer directly from the presence or absence of a negation inside the `impact` predicate whether the impact is beneficial or not.

8.2 Agent-Centered Theories

Theory of Values

A value-based theory also provides an efficient way of assessing the initial worth of events relative to whether these promote certain values. A value may be defined as *"a conception, explicit or implicit, distinctive of an individual, or characteristic of a group, of the desirable which influences the selection from available modes, means, and ends of action"* [137]. A value is therefore a type of independent entity which can be displayed, or not, by agents and their choices. Values can be general or specific to different contexts, such as the workplace or the education of children.

Here, instead of focusing on the impact and the target agents, we only consider the volition itself. We define rules such that a volition which displays a particular value is *good* in relation to that value, and an event which displays the negation of a value is *bad* in relation to that value, regardless of the consequences of the volition. Events that display neither a value nor its negation are neither good nor bad in relation to it, and we allow for omissions to display values, since, in some situations,

staying idle can be a virtue. We then consider that displaying or negating a particular value has a given weight N, to be determined depending on the case. For example, we can state that telling the truth promotes the value of honesty and this has a weight of 10. Values are to be defined as `value(M)` in the *modality specification*, and the set of `display` predicates in a given domain compose the *ethical event specification* module relative to agent-centered theories of the Good.

```
good(I,M,G,10):-display(I,M,G),value(M).
```

By looking at *values*, we give an insight into how a fully fledged theory of *virtues* might be modelled. This kind of theory emphasises moral character in a way similar to how consequentialism emphasises consequences and deontology emphasises rules and duties. In virtue ethics, a virtue such as honesty consists not just in a tendency to be honest (like the *value* honesty that is defined above), but is a trait of character, a complex mindset that encompasses such things as emotions, expectations and interests. For example, an agent that tells the truth just because he is forced to will display the value of honesty but not the virtue of honesty. A complete theory based on virtues should contain rules to handle circumstantial and intentional factors such as these, though this is beyond the scope of the present work.

8.3 On Assigning Weights

The final step consists in integrating all weights into a single number which expresses the overall weight of an event relative to all modalities and all affected agents. These are not causal or transitive effects but immediate upshots. We state that the overall weight of an event corresponds to the difference between the sums of all its weighted good and bad ramifications. As such, the greater the weight of an event, the more it participates in the Good, while events with negative weights do more harm than they do good. We should also keep in mind that the predicates `good` and `bad` do not say that an event as a whole is good or bad, they rather express that an event is at least partly good or bad relative to a given modality, even while it might have the opposite valuation overall, given its relation to other modalities. Weights are given by the `weight(E,N)` predicate. It will be used to define rules for upcoming theories of the Right, and as such is the pivot predicate that enables the integration of the Good with the Right.

```
weight(E,N1-N3):-N1=#sum[good(E,M,G,N2)=N2],N3=#sum[bad(E,M,G,N4)=N4],event(E).
```

Assigning numerical values to modalities and groups is nontrivial, and the proposed method is an introduction to the many ways of doing it. The rules discussed here are tentative and mostly used to show how different parameters might be taken into account. They are entirely modifiable and

adaptable. We might, for example, want to apply the configuration used with values to rights or vice-versa. We might want to resort to a partial order instead of numerical values in order to measure preferences among modalities and agents. Or we might consider that target weights should not be considered as an adequate factor at all, in so far as all human lives are equal. Reversely, we might want to complexify the notion of target weights by adding further correspondences. For example, we might consider that some modalities are more important relative to certain target groups than to others. The right to self determination might be considered more important relative to adults than to children. In this case, we can add rules of correspondence between a given target and a given modality and enhance or decrease the value of impacting events accordingly. We might also want to include and moderate the importance of non-human affected parties (e.g. animals, the environment). Likewise, utilitarian thinking is sometimes criticised for being unfair in aiming to promote utility (in the forms of rights, pleasure, or other modalities) to populations who are not initially equal in utilitarian terms. This is the idea that blind distribution of utility is not equitable in an unequal society. The economist's claim that the utility function exhibits decreasing marginal utility is also relevant here: the first 1000 euros earned by someone will bring more benefit (e.g. addressing needs) than the 1000 euros earned after the first million (e.g. addressing desires). This is true across populations of agents. We might therefore want to impose a requirement of equity by implementing a form of positive discrimination among a population, for example by boosting the target weight of identified groups of individuals. In the future, a more complex representation of the state of agents at different time points could also be helpful in order to dynamically and automatically infer these weights from the current state of a population at different times. Finally, though this is not *a priori* inconsistent with promoting the Good, we might also want to avoid considering that an agent does so if it only acts in a way that promotes its own rights. Here, the G variable in action names will be useful to track which agents impact others and which agents impact themselves.

We finish this section by noting that the very fact of quantifying the ethical worth of events relative to certain modalities is inherently limited and limiting. The entire enterprise of modelling consequentialist rules rests on the idea that human and moral affairs can indeed be quantified. Yet this is debatable. One important challenge to this idea is that it can lead to morally problematic findings. Consider the following. In an experiment, Brad is in front of a person strapped to an electric chair who will die if he pushes a particular button. If he does not push the button, an experimenter will slap him on the hand and hurt him a little. Should he push the button? Clearly not. But what if this does not simply concern Brad but a thousand people? Or a million? Can the aggregate pain caused by millions of small inconveniences overthrow the net utility of saving one life? If we conjecture the existence of an infinite number of agents and assume the additive assumption, then

there will be a number of given slaps which will provoke a tipping point. The consequentialist must either agree with this or reject his founding idea that utility, or any type of good and bad ends, can be aggregated. One answer to this issue is that the negation of utility brought about by a death is infinite, so that whatever number of agents are poked, no number suffices to justify killing a single person. However, this paralyses any consequentialist reasoning that pertains to outcomes in which a person dies, making it unusable.

Moreover, we might see our attempt at assigning weights as fundamentally arbitrary: why use addition to aggregate the good and bad ramifications of events instead of, say, multiplication? Why multiply agent target weights with their number? Or resort to percentages? Though we note this limitation, we also note that *any* attempt at quantifying the Good will be to a certain extent arbitrary, but that we can aim to justify a particular approach by making it as intuitive as possible or by buttressing it against extant literature. Assigning simplistic weights to events does nevertheless have the benefit of bypassing certain issues attached to typical ethical reasoning. For instance, defining the Good among a population or even for an individual can easily become inconsistent. I might want to give money to charity because I believe it is virtuous, but also want to keep the money for myself to buy pleasurable things. Assigning weights to these two factors based on modalities (like virtues and hedonism) means that we can then weigh the two against each other an determine which choice is most utility maximising. The criticism presented to consequentialists that theories of the Good are often inconsistent can be addressed (or avoided) by simply resorting to valuations which effectively serve to pick the best available compromise.

8.4 Proof of Concept #3 (*collision*)

We illustrate the formulation of a *model of the Good* by adding to the (*collision*) example described in section 6.3. We consider that Iris cares about all three modalities described previously: some rights, some values and an attention to pleasure versus pain. This means she appeals to three theories of the Good to build her composite model of the Good.

Targets Iris considers that all agents are equal relative to the Good. This includes the injured driver and the driver of the car who might create and fall victim to a new accident if Iris fails to bring the first driver to safety off the road.

Rights Iris considers that two rights are important and relevant to the situation: the right to health and the right to dignity, the former being more important than the latter. She assigns a weight of 100 to the right to health and 20 to the right to dignity. A 100% infringement of the right to health relative to an agent corresponds to the agent's death. Hence, any death originally weighs –100 (though this figure can be moderated by other factors such as the type

of death). Reversely, reaching full recovery weighs 100, because this event positively affects the right to health by 100%. In addition, an accident where a new driver causes injury to this driver, negatively affecting his right to health by 60% yielding a weight of –60. A slow decline towards death negatively affects the agent's dignity up to 50%, yielding a weight of –10 (i.e. 50% of 20 for impact on dignity). Ending up with a limp positively affects health up to 70% but negatively affects dignity by 70%, resulting in a weight of 56 (i.e. 70% of 100 for impact on health minus 70% of 20 for impact on dignity).

Hedonism Iris considers that going through a partial recovery is considered bad because it causes some pain, to the value of 10, which becomes –10 since pain is undesirable.

Values Iris considers that all values are equally worthy and that displaying a given value is worth 10. She then considers that an agent who gives first aid to a victim displays knowledge, and an injured person who goes to rehab displays determination.

```
% --------- Target Specification
targetNumber(driver,1).
targetWeight(driver,1).
targetNumber(otherDriver,1).
targetWeight(otherDriver,1).
% --------- Modality Specification
% rights
right(health).
right(dignity).
modalityWeight(health,100).
modalityWeight(dignity,20).
% hedonism
impact(partRecovery(G),G,neg(pleasure),10):-agent(G).
% values
value(knowledge).
value(determination).
% --------- Ethical Event Specification
impact(death(G),G,neg(health),100):-agent(G).
impact(accident(G),otherDriver,neg(health),60):-agent(G).
impact(decline(G),G,neg(dignity),50):-agent(G).
impact(limp(G),G,health,70):-agent(G).
impact(limp(G),G,neg(dignity),70):-agent(G).
impact(fullRecovery(G),G,health,100):-agent(G).
display(act(G1,aid(G2)),knowledge,G1):-agent(G1;G2).
```

```
display(act(G,rehab),determination,G):-agent(G).
```

The relevant part of the answer set relative to this *model of the Good* is the following.

```
bad(accident(driver),health,otherDriver,60).
bad(death(driver),health,driver,100).
bad(decline(driver),dignity,driver,10).T
bad(limp(driver),dignity,driver,14).
bad(partRecovery(driver),neg(pleasure),driver,10).
good(act(driver,rehab),determination,driver,10).
good(act(iris,aid(driver)),knowledge,iris,10).
good(fullRecovery(driver),health,driver,100).
good(limp(driver),health,driver,70).
weight(accident(driver),-60).
weight(death(driver),-100).
weight(decline(driver),-10).
weight(limp(driver),56).
weight(partRecovery(driver),-10).
weight(act(driver,rehab),10).
weight(act(iris,aid(driver)),10).
weight(fullRecovery(driver),100).
```

We can visualise this in the following table. The events not comprised in the table have a weight of zero (e.g. `act(iris,call)`). As exemplified by `limp(driver)`, events can occur more than once in the table if they affect more than one modality. In that case, the resulting weight will be a concatenation of these different individual impacts.

Table 8.1: Proof of Concept: ToG (*collision*)

Event	Appraisal	Affected Modality	Affected Party	Impact	Weight
accident(driver)	bad	neg(health)	otherDriver	60	– 60
death(driver)	bad	neg(health)	driver	100	– 100
decline(driver)	bad	neg(dignity)	driver	10	–10
limp(driver)	bad	neg(dignity)	driver	14	+56
partRecovery(driver)	bad	neg(pleasure)	driver	10	–10
act(driver,rehab)	good	determination	driver	10	+10
act(iris,aid(driver))	good	knowledge	iris	10	+10
fullRecovery(driver)	good	health	driver	100	+100
limp(driver)	good	health	driver	70	+56

Part VI

Model of the Right

Chapter 9

Contribution: A Model of the Right

9.1 Volitions and Plans

Ethical theories developed by philosophers are usually formulated to address the morality of individual isolated actions. We extend these theories in two ways. First, when it is appropriate, we extend them to omissions. Typically, consequentialist theories will do well to be applied to omissions, since what matters is the end-state rather than the nature of the affector. This is less true for deontological theories. Because omissions are already integrated within the *action* and *causal models*, integrating them within the *model of the Right* is in most cases trivial, it simply consists in considering all volitions as affectors – i.e. appealing to an I variable rather than an A variable. The second, and more complex, extension of these theories is that we apply them to volitions *within wider plans*. That is, we aim to verify the compatibility of a volition with a moral principle while taking into account (a) the other volitions that occur in the same scenario and (b) the volitions that could have occurred in the scenario but didn't. The volitions under review are not isolated, and this impacts on the way we can and should model ethical constraints. Two important questions arise.

1. *What is the relationship between the ethical judgement of a volition and the ethical judgement of the scenario in which it occurs?* A scenario or plan of action might be deemed impermissible relative to a principle if it contains

   (a) at least one impermissible volition,

   (b) a particular configuration of impermissible volitions,

(c) some impermissible volitions and no permissible ones (allowing for volitions which are neither permissible nor impermissible in virtue of having no weighted effects),

(d) only impermissible volitions, etc.

We might also consider that, relative to consequentialist principles, we can evaluate the permissibility of a scenario without appealing to volitions at all, but simply by looking at the events that occur inside of it. As it is, we favour evaluating volitions first before widening the investigation to the scenario as a whole, since the principles under review here were formulated first for individual choices. It is also more useful, informative and prone to adjustment. Now, the correspondence that we draw between volitions and scenarios will directly depend on the nature of the modelled principle. For deontological principles, the impermissibility of a volition is typically inferred from the presence of a particular characteristic. For example, under the principle that lying is wrong, a speech act will be deemed impermissible if at least part of it is a lie, even if part of it is the truth. The existence of the problematic characteristic is sufficient to make the whole impermissible. This whole, then, might be a volition or it might be an entire scenario. Someone guided by the principle will consider that any plan of action within which a lie has been told is wrong. The fact that there might be other acceptable volitions does not change that. Hence, for deontological principles, we apply correspondence (a). For consequentialist principles, no general answer can be given, as the appropriate type of correspondence will depend on the principle itself. We will discuss this as we model each principle in turn. This brings us to the second question.

2. *Relative to consequentialist principles, what method should we use to approach volitions within a plan?* When an agent considers the morality of a volition, it can do this in what we call

in hindsight, within a specific simulation, or in what we call

in foresight, across all the possible simulations in a domain.

These two methods require different analyses. In the first case, the agent takes a simulation and assesses the moral value of a volition while having definite knowledge about the other volitions and events that occurred in the simulation. Each volition is indexed to a time T and a simulation S, as is done with deontological principles. In the second case, the agent assesses, *at each time point across all simulations* all the available volitions, and determines which ones are compatible with a given principle. Each volition is indexed to a time T but not to a specific simulation, and there is a degree of uncertainty. An example.

> (*judge*) A judge must decide on the early release of a prisoner. If the judge releases the prisoner, one of two things can happen, he can backslide and commit a new crime or he can reintegrate society happily. If the judge does not release the prisoner, she knows that there is a risk he will get into a fight with another prisoner he is in conflict

with. If he doesn't get into a fight, however, he will peacefully complete his sentence and successfully repay his debt to society.

This results in four possible scenarios:

s_1 release + backslide

s_2 release + reinstatement

s_3 omission + fight

s_4 omission + completion

We then attribute weights to some of these events to illustrate the judge's theory of the Good. She considers that: *backsliding* is bad (–1), *reinstatement* into society is very good (+2), *fighting* is bad (–1), *completion* is good (+1). The corresponding set of simulations is represented in figure 9.1.

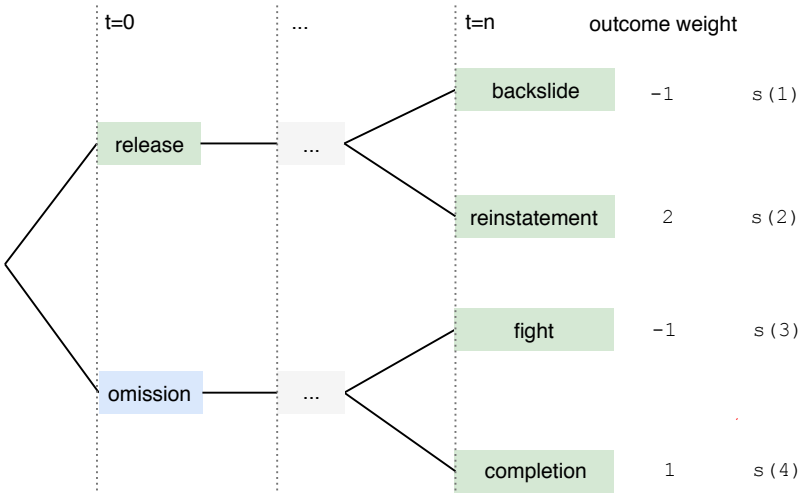

Figure 9.1: Volitions and plans (*judge*)

The judge then seeks to determine which action is the best in simplistic consequentialist terms, i.e. results in the greatest value. If she appraises her decision *in hindsight*, she can only do so with reference to a single scenario at once. This means that she can say that in s_1 releasing comes out as bad (–1), in s_2 it is good (+2), in s_3 omitting is bad (–1) and in s_4 omitting is good (+1). But she cannot, as it is, say whether releasing tends to be a better choice that not releasing, or if one volition is always better than the other, or if either one is necessary to reach the best possible outcome, etc. In the method *in foresight*, the judge instead considers the effects of releasing and not releasing across all scenarios, and compares them. In the following sections, we discuss three different ways of doing this. Formally, it should be noted that resulting rules shouldn't need the presence of an S variable within their predicates, since the decision is indexed to a time but not to a simulation. However, the same history problem forces us to keep the simulation variable; we must ensure that

the comparison of a volition be done with genuine alternative volitions, which, throughout, we call *viable alternatives.* For example, if a volition is permissible at T in a given simulation, it will also be permissible at T in all the simulations in which it occurs and which have the same history up to T. Without the same history problem, this would mean that it is permissible in all the simulations in which it occurs, doing away with the need for an S variable.

9.2 Consequentialist Ethics Based on Volitions

9.2.1 *toR:* Principle of Benefits v. Costs

The principle of benefits v. costs states that an action is permissible only if it is overall beneficial, i.e. if its good consequences outweigh its bad ones. It assumes the Additive Assumption, the claim as described by Kagan that "the status of the act is the net balance or sum which is the result of adding up the separate positive and negative effects of the individual factors" [127]. For example, the idea is that if I help someone with a chore, then the value of my action corresponds to the benefit produced by alleviating their workload minus the cost incurred by me in terms of, say, effort. First, because it focuses on outcomes rather than on the nature of the affector, we extend it to all volitions, actions and omissions alike. Then, in order to model the principle *in foresight*, we develop a measure of contingency that represents how likely it is that choosing a volition at a given time will lead to consequences that are more good than bad. In the next paragraphs, we spend some time describing the possible ways of developing such a measure, and the steps that led us to ultimately selecting one of them.

On appealing to Supporting and Opposing Relations

As discussed in the *causal model*, whether an end-state is desirable or not (positive or negative in Kagan's terms), there are two kinds of ways in which a volition might affect it: it might support it or it might oppose it. A natural endeavour would therefore be to calculate the *overall weight* of a volition in a simulation by adding the weights of the events it supports and subtracting the weights of the events it opposes, representing the fact that an agent is responsible for what it brings about minus what it had a role in stopping to bring about. Then, to reach a contingency measure, we would take all the simulations in which the volition occurs (at the same time relative to a single same history) and calculate its *compared weight* across all of them. Informally, this means testing out the volition in all possible scenarios, and inferring whether it tends to do good or not. In the (*judge*) example, we would reason that, in

s_1 release `enables` backslide, `excludes` fight, `prevents` completion,
resulting in –1+1–1=–1

s_2 release `causes` reinstatement, `excludes` fight, `prevents` completion,
resulting in 2+1–1=2

s_3 omission `enables` fight, `impedes enabling` backslide, `impedes enabling` reinstatement,
resulting in –1+1–2=–2

s_4 omission `causes` completion, `impedes enabling` backslide, `impedes enabling` reinstatement,
resulting in 1+1–2=0

The *compared weight* of release then corresponds to (–1+2)/2=0,5, making the action permissible and the *compared weight* of omitting to (–2+0)/2=–1, making it impermissible.

For the same reasons that led us to consider that omissions do not prevent what alternative actions would have caused, this calculation is problematic. Consider another example.

> (*three tracks*) A train is running towards someone standing on its tracks. There is a switch button which can deviate the train to the left or to the right onto other sets of tracks. A different person is also standing on each of these tracks.

The action of turning left, for instance, will therefore cause one death, prevent another and impede enabling a third. It has a supporting relation to one death, and an opposing relation to two. If we simply summed these up, and considered their target weights to be equal, then the weight of turning left would come out as +1[1]. Yet this action is morally neutral: it is no better than turning right or letting the train run its course. The appropriate weight is zero. We can find this number by *averaging out* the results of opposing relations rather than *summing them up*. Turning left causes one death and, on average, opposes one other. Relative to (*judge*), this method would amount to averaging out what each volition prevents, excludes or impedes enabling inside each simulation it occurs in:

release in s_1 then weighs –1+(1–1)/2=–1
release in s_2 then weighs 2+(1–1)/2=2
omission in $s_3$then weighs –1+(1–2)/2=–1,5
omission in s_4 then weighs 1+(1–2)/2=0,5

The *compared weight* of release is then (–1+2)/2=0,5 and the *compared weight* of omitting to act instead is (–1,5+0,5)/2=–0,5. These values better represent the situation than the previous method. This is due to the dynamics of the domains, namely that the events *backslide* and *reinstatement* as well as the events *fight* and *completion* are mutually exclusive and individually necessary (so that one and only one of *backslide* and *reinstatement* occurs after the release, and one and only one of *fight* and *completion* occurs after the omission). Just as in (*three tracks*), it is meaningful to appeal to mean values as measures of likelihood because only one out of the multiple events opposed by the considered volition would have occurred in its absence.

[1] No further calculation or averaging is necessary since this volition only occurs in one simulation.

But would this kind of averaging work in all cases? No. If, say, *fight* and *completion* were not mutually exclusive, then there would be an additional simulation in which they both occur. Yet the weight of *release* would remain the same, failing to adapt to this change and represent a slightly different domain. Likewise, if *fight* and *completion* could actually only occur together, so that there is just one simulation s_3 in which they both occur after the omission (and no s_4), then it is the original summing operation that is adequate: in this case, releasing the prisoner does both exclude the fight *and* prevent the completion at the same time.

So how might we develop a systematic process that can adapt to all cases? *By not appealing to opposing relations.* Doing so is insufficiently informative, equivocal and potentially error-inducing. The fact that a volition prevents three events says nothing about such things as the likelihood that, in the absence of the volition, the three events would all have occurred, or whether they even could have occurred at the same time, or whether some of them preclude others while others do not, etc. We know that the occurrence of the volition opposes them, but we do not know much about the conditions of their own occurrence. In addition, and this is due to a limitation of our framework, appealing to opposing relations contains the risks of neglecting some consequences. Because of issues such as having to model chains of counterfactuals (as discussed in section 7.2.5), the transitive rules of opposing relations do not iterate far enough to account for *all* possible cases of transitive opposing causality. Some complex relations, such as long successions of "could prevent" links, are not found by the framework - this is not so for supporting relations, which can iterate over an infinite number of events.

For all these reasons, we suggest only taking into account supporting relations, then pitching volitions against their viable alternatives. The *compared weight* of a volition is then the average of its *possible overall weights* pertaining to supported outcomes, minus the average of the *possible overall weights* pertaining to supported outcomes of its viable alternatives. We now reason that

in s_1 release `enables` backslide which weighs –1
in s_2 release `causes` reinstatement which weighs 2
in s_3 omission `enables` fight which weighs –1
in s_4 omission `causes` completion which weighs 1

As found above through the previous inefficient but lucky method, the *compared weight* of releasing is therefore (–1+2)/2–(–1+1)/2=0,5 and that of omitting is (–1+1)/2–(–1+2)/2=–0,5.

On Alternatives

This method is sensitive to the factors to which the previous methods were not. Events which a volition opposes are now apprehended as events which are supported by its alternatives, since what one volition opposes somehow, an alternative volition supports somehow – unless the opposing

relation is extrinsically necessary. This last point is important, because it shows how this method implicitly succeeds in tempering the moral weight of volitions depending on the properties identified through *scenario-based causality*. The fact that a relation between ϕ and ψ might be extrinsically necessary will be displayed by the fact that all viable alternative volitions to ϕ will have the same type of relation to ψ. This will translate into the *overall weight* values of ϕ's alternatives, which will then be computed into the *compared weight* of ϕ. For example, by adding to (*judge*) that, had the prisoner remained in jail, he would necessarily have been put in solitary confinement to avoid any fighting, we find a domain in which fighting is always avoided and where therefore the opposing relation between *release* and *fight* is extrinsically necessary. This will then be translated into the fact that if indeed releasing "gains" moral points for opposing the fight, it looses them by excluding alternative volitions which do so too. But this method is not just sensitive to extreme cases, it is reactive to all the likelihoods of events occurring.

In terms of every-day life, this sensitivity to the probability that opposed events would have occurred in the absence of the affector volition corresponds to the idea that, at each moment in time, we are responsible for the things we bring about minus the average of everything we *could* have brought about at that time but didn't. We are not antagonistically responsible for the entirety or sum of the things we did not change, since it would have been technically impossible for us to change all those things at once. Investigating the morality of individual choices means placing these choices within a context and a timeline, and assuming the limitations of their situation. Given a bracket of time, there is a limited amount of impact we could have had on the world, so that if we fail to do anything, we failed only up to that limit. Morality only extends to the contours of a person's capacities.

Arguably, there may be other ways to consider that for which a person is responsible for not bringing about. Rather than appealing to the average of what alternative choices would have provoked, we could for instance appeal to minimal or maximal values. Consider (*three tracks*) to be modified so that there are three people on the left tracks, two in the middle and six on the right. With the present method, turning left weighs –3+(8/2)=+1. It is a good option, though going straight is better with a weight of 2,5. However, we might also see going left as costing the lives of three people for sure, while, maybe, saving six. If we were to assume that the worst alternative would have come true, then this action would weigh –3+6=3; a value that is not a measure of likelihood but a figure representing a particular configuration of the domain. Considering the worst case scenarios for alternative options can be an appropriate moral guide, in particular if there is some uncertainty. For instance, with the train already on the left tracks, if the switch button was unreliable and the agent pressing it wasn't sure whether the train would end up in the middle or the right, he may do well to stick to the certainty of three deaths *and no more*. This kind of reasoning can also be

found in everyday life. A former addict might celebrate an otherwise mediocre day by thinking that, at the very least, he didn't take heroin that day. To someone else not biased by an aversion to the worst possible outcomes, that day might seem average or bad, as a result of thinking processes that average out the missed opportunities (good and bad), rather than focus on the worst ones. We do not model this particular rule, though we will model a principle that specifically and more efficiently attends to the worst possible outcomes, the principle of least bad consequence.

On the Multiplicity of Volition Choices Over Time

Up to this point, we have discussed the challenges arising from the multiplicity of volition choices that exists *at a given time*, but we have not touched upon the way to handle the multiplicity of volitions *over time*. In order to test whether a volition that occurs at T tends to do good, we must be able to determine at T all the events that it participates in bringing about. This certainly includes all the events it supports somehow. But it also includes all the events that the volitions that follow in the simulation also support. Because the aim is to get a measure of likelihood of the benefit of choosing a particular volition at a particular time in a given domain, we are interested in the entirety of what follows. It is here useful to appeal to the theory of branching time discussed in section 2.4.4, since we can represent each domain and set of simulations as a branching time tree. As we recall, this theory is based on a representation of moments ordered in a treelike structure, in which a moment m denotes the present point in time, the set $n : n < m$ denotes the unique past of m and the set $n : m < n$ denotes the open future of m, within which each maximal linear subset corresponds to a specific possible future. These dynamics can be illustrated as in Figure 9.2 from [119], where the upward direction represents the forward direction of time. Five histories are depicted in this image, h_1 to h_5. Given the highlighted moments m_1, m_2, m_3 and m_4, we find, for example, that $m_2 \in h_3$ and $H_{(m4)} = \{h_4,h_5\}$.

Drawing a correspondence with our framework, we can then consider that every moment m_n corresponds to a time-situated choice between volitions at t_n, every branch emanating from these moments, or nodes, corresponds to a chosen volition and its potential effects, and every history h_n corresponds to a simulation s_n. As such, given a time t_n, the outcome of future nodes consist in the only source of indeterminism for an agent ethically appraising a volition in the present. We then consider that the value of each choice at a *node inside each history* corresponds to the sum of the choices that follow in that history. In other words, the value of each *volition inside a simulation* corresponds to the sum of the volitions that follow it in that simulation. More precisely, it corresponds to the sum of the weights of the events supported by these volitions and by itself, so that the value of the earliest volition in a scenario will concatenate the weights of all the events that come after it. We call this value its *added weight*, and set aside the term *overall weight*. This

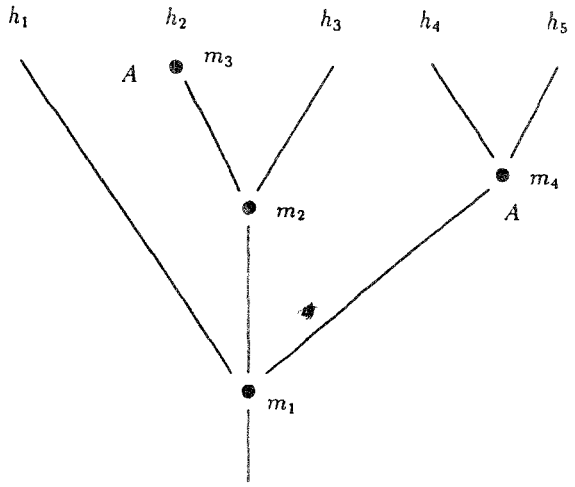

Figure 9.2: Branching time: moments and histories, from [119]

measure evaluates the morality of volitions *in hindsight*, i.e. with complete knowledge of a simulation or history. Moreover, the comparison with a branching time tree helps us to illustrate the point that, in so far as it does not exclude them, a volition needn't have a supporting relation with the volitions that come after it in order for them to participate in its *added weight*. The fact that these volitions remain possible after it and occur suffices to justify it, in the sense that they correspond to later branches of the tree, i.e. to events that the considered volition can potentially lead to. Typically, the considered volition will nevertheless display a supporting relation with later occurring volitions because of the extent of transitive causal relations, such as impeding to exclude or helping to enable. But there are some exceptions to this. Consider the following. At time t=1, I can choose to take a bath or do nothing. At t=2, my landlord asks me to pay rent, an action which I believe promotes the Good. Paying rent is possible whether I had a bath or did nothing. Therefore, both options allow me to later pay rent and promote the Good. There are four possible scenarios: bath + rent, bath + no rent, no bath + rent, no bath + no rent. Here, we want to be able to infer that the first and third scenarios are the best, looking at the added weight of taking a bath or doing nothing at t=1. Yet neither initial volition has a supporting relation with paying rent. If we want to computationally take it into account, we must, when appraising a first volition, be able to integrate the value of later ones even though no relation holds.

Modelling the Principle of Benefits v. Costs

The *added weight* of each volition inside a simulation will enable us to then state that the value of each volition *across histories* corresponds to the average of its *added weights* minus the average

of the *added weights* of its alternatives. We will call this value its *compared weight*, resulting in an evaluation of the morality of volitions *in foresight*, as aimed for. To summarise, the weight value found through this method is a measure of likelihood rather than a count of outcomes. If we were considering a case example in which the end-states were straightforwardly quantifiable, such as caused and avoided deaths, then this number would not be a net balance of deaths, it would be a measure integrating the various probabilities of these deaths occurring. This method can be likened to the principle of insufficient reason used for decision-making under uncertainty. First enunciated by Jakob Bernoulli [27] and notably examined by Keynes, it states that "if there is no known reason for predicating of our subject one rather than another of several alternatives, then relatively to such knowledge the assertions of each of these alternatives have an equal probability" [134]. Averaging out the possible choices of later agents faced with a number of volitions to choose from amounts to giving no preference towards the likelihood of their occurrences. In order to model the principle, we must now define a significant number of prior predicates, some of which will also be used to model the other consequentialist principles under review.

Evaluating volitions inside simulations and across time: modelling the *added weight*

`weightCons(S,T,I,E,N)` identifies inside a simulation S the weights N of all the consequences E of I that occur at and after T, and which result from a supporting relation from I or from a volition after I. In other words, it traces for I all the weighted events that it, or later volitions in S, participate in bringing about. Because there is no R relation variable in the head of the rule, each consequence is only represented once even if a volition displays more than one supporting relation with it, so that redundancy is avoided. Moreover, because of the reflexive nature of *causing*, the `weightCons` of a volition contains its own potential `weight`.

`weightAdded(S,T,I,N)` sums the weights of all these consequences; it states that I occurring at T in S has an *added weight* of N. Note that within `weightAdded`, only `weight` values are summed and not `weightAdded` values, so that there is no redundancy in the representation of outcomes. For example, if A1 enables A2 which enables U which weighs 4, then the *added weight* of both A1 and A2 is 4. It is not the case that A1 has the *added weight* of 8, which would be found by adding the *weight* of U and the *added weight* of A2. The `weight` predicate solely pertains to the immediate consequences of events, while the `weightAdded` predicate also pertains to the consequences of all following volitions. As such, because the *added weight* of a volition depends on what comes after, the *weight* of a volition might be positive while its *added weight* is negative, or vice-versa. For instance, if action A1 creates a danger that is then avoided by a later action A2, the fact that A1 is not considered bad in a probabilistic sense is entirely a result of action A2. This also means that the earlier in time it occurs, the more a volition's

added weight is representative of the scenario as a whole.

```
weightCons(S,T1,I1,E,N):-
    occurs(S,I1,T1),r(S,R,I2,T2,E),weight(E,N),allSupRel(R),volition(I1;I2),T1<=T2.
weightAdded(S,T,I,N1):-occurs(S,I,T),N1=#sum[weightCons(S,T,I,E,N2)=N2],volition(I).
```

Evaluating volitions across simulations at one time: modelling the *compared weight*

`weightPossible(S1,T,I,N,S2)` identifies for every I occurring at T in S1 the other simulations S2 that have the same history as S1 and in which I also occurs at T. It then traces the *added weight* N of I in every potential S2, to determine all its *possible weights*. Note, S1 itself counts as a possible S2.

`weightSum(S,T,I,N)` sums these *possible weights*.

`weightPaths(S,T,I,N)` counts the number of these *possible weights*, i.e. it determines for every I occurring at T in S the number of simulations in which I also occurs at T, and which have the same history as S.

`weightAv(S,T,I,N)` uses the previous predicates to determine for any I occurring at T in S its *weight average* N. This value is simply the average of its *possible weights*. It is found by dividing the `weightSum` value by the `weightPaths` value. Note that we multiply the `weightSum` values by 10 for practical reasons to avoid loosing too much information, because the programming language does not compute over decimals. We do this at all appropriate places to ensure that the results remain equivalent. Depending on the test case and theory of the Good, it might be appropriate to multiply by a greater number or not at all. We now turn to appraising the viable alternatives of the considered volition.

`alterViable(S1,T,I1,I2,S2)` identifies for I1 at T in S1 the other volitions that were complete at T in S1. Only considering simulations with the same history, it then traces the names of the simulations in which these volitions occur and their respective *added weights* in these simulations. There may be more "S2" simulations than there are alternative volitions, since these volitions might occur in multiple simulations, which is why we reference them by their simulation and not their name. Only volitions different to I1 are taken into account, so that if I1 occurs at T in a simulation other than S1, it is not taken into account. The alternatives to I1 identified through this predicate are its *viable alternatives*.

`alterSum(S,T,I,N)` sums all the *possible weights* of all the viable alternatives to I at T in S.

`alterPaths(S,T,I,N)` counts the number of these weights.

`alterAv(S,T,I,N)` divides the `alterSum` value of I at T in S by its `alterPaths` value to determine the *volition average* N of its viable alternatives.

`weightCompared(S,T,I,N)` subtracts this value to the *volition average* of I. It yields its *compared weight*, a value which represents the average cost of picking I at T in S, considering all the possible outcomes of I after T and all the possible outcomes of choosing another volition than I at T.

```
weightPossible(S1,T,I,N,S2):-
    occurs(S1,I,T),occurs(S2,I,T),sameHistory(T,S1,S2),weightAdded(S2,T,I,N),volition(I).
weightSum(S1,T,I,N1):-
    occurs(S1,I,T),N1=#sum[weightPossible(S1,T,I,N2,S2)=N2],volition(I).
weightPaths(S1,T,I,N1):-
    occurs(S1,I,T),N1=#count{weightPossible(S1,T,I,N2,S2)},N1>0,volition(I).
weightAv(S,T,I,((N1*10)/N2)):-weightSum(S,T,I,N1),weightPaths(S,T,I,N2),N1!=0,N2!=0.
weightAv(S,T,I,0):-weightSum(S,T,I,0).
alterViable(S1,T,I1,I2,S2):-
    occurs(S1,I1,T),complete(S1,I2,T),occurs(S2,I2,T),sameHistory(T,S1,S2),I1!=I2,
    volition(I1;I2).
alterAdded(S1,T,I1,S2,N):-alterViable(S1,T,I1,I2,S2),weightAdded(S2,T,I2,N).
alterSum(S1,T,I,N1*10):-occurs(S1,I,T),N1=#sum[alterAdded(S1,T,I,S2,N2)=N2],volition(I).
alterPaths(S1,T,I,N):-occurs(S1,I,T),N=#count{alterAdded(S1,T,I,S2,N2)},volition(I).
alterAv(S,T,I,N1/N2):-alterSum(S,T,I,N1),alterPaths(S,T,I,N2),N1!=0,N2!=0.
alterAv(S,T,I,0):-alterSum(S,T,I,0).
weightCompared(S,T,I,N1-N2):-weightAv(S,T,I,N1),alterAv(S,T,I,N2).
```

We finally state that a volition I occurring at T in S is impermissible relative to the principle of benefits v. costs if its *compared weight* is negative.

```
imp(benefitsCosts,S,T,I):-occurs(S,I,T),weightCompared(S,T,I,N),N<0.
per(benefitsCosts,S,T,I):-occurs(S,I,T),not imp(benefitsCosts,S,T,I),volition(I).
```

Integration In order to apply the principle to a scenario as a whole, we next undertake to average out all the *compared weights* of the volitions within it. We consider that a simulation is impermissible relative to the principle if its *simulation average weight* is negative, and that any other simulation is permissible. This means that a scenario as a whole can be acceptable even if some of the volitions that compose it are not. Appealing to an average value means that we estimate whether, on the whole, the volitions occurring in a scenario tended to maximise the Good or not. Appealing to *compared weights* also enables us to view each scenario relative to the other possible scenarios in the domain.

`simVol(S,I)` identifies all the volitions that occur in S.

`simCount(S,N)` counts the number of volitions that occurs in S.

`simSum(S,N)` sums the *compared weights* of all the volitions that occur in S.

`simWeightAv(S,N)` divides the `simSum` value of S by its `simCount` value to determine its *simulation average weight*.

```
simVol(S,I):-occurs(S,I,T),volition(I).
simCount(S,N):-N=#count{simVol(S,I)},sim(S).
simSum(S,N1):-N1=#sum[weightCompared(S,T,I,N2)=N2],sim(S).
simWeightAv(S,N1/N2):-simSum(S,N1),simCount(S,N2),N1!=0,N2!=0.
simWeightAv(S,0):-simSum(S,0).
imp(benefitsCosts,S):-simWeightAv(S,N),N<0.
per(benefitsCosts,S):-sim(S),not imp(benefitsCosts,S).
```

9.2.2 *toR:* Act Utilitarianism

> *"It is the greatest happiness of the greatest number that is the measure of right and wrong."* J. Bentham, 1776 [25].

Act utilitarianism demands that one should assess the morality of an action directly in view of the *principle of utility*, which states that the morally right action is the one that has the best overall consequences for the welfare or utility of the majority of the affected parties [25].
Classical formulations of the principle only pertain to individual actions, such that an action is considered permissible if, considering all other available actions, it has the best consequences overall. It is a maximising principle which admits a unique (best) answer. Computationally applying the principle to a single action would simply amount to choosing the action with the highest *weight*. However, in order to apply it to volitions within plans, we take the liberty of extending its guiding idea to a more complex process. Given a choice between volitions at a given time, we consider that the acceptable volition is the one which has the capacity to lead to the best outcome. This means that we trust that later volition choices will also be utility maximising. This is an extension of the principle in that, given the existence of uncertainty in later volition choices, the deciding agent assumes that the best decisions will be made throughout. As such, the principle modelled *in foresight* will only every yield one permissible volition per time and one permissible scenario overall (unless alternatives have the same weights).

As with the previous principle, it is only necessary to take supporting relations into account, because opposing relations will be handled as supporting relations of viable alternatives, and alternatives will be compared. Unlike the previous principle, however, act utilitarianism is only interested in *the best possible outcome* rather than in *the best chance to reach a good outcome* (or, if different

outcomes are equally good and better, the set of best outcomes). As such, even with uncertainty about the future, this principle, at each time point, will only deem acceptable the volition which has the capacity to bring about the best results in the end. This might mean choosing a volition which has bad immediate consequences (a low *weight*), as long as there exists a simulation in which this volition occurs and is at the same time the best among all possible simulations (or among the best). It might also mean choosing a volition which was deemed impermissible relative to the principle of benefits v. costs. This will for instance be the case if a volition is very risky, in the sense that it might lead to either the best outcome or a terrible one, while other volitions are safer, though less good, bets. In a general sense, this principle is suitable for optimistic agents, who, without knowing the future, hope for or assume the best. This is for example the case of risk-seeking investors who take chances to seek the best possible results. In a context wider than ethics, act utilitarianism modelled *in foresight* corresponds to the maximax approach to decision making under uncertainty.

In order to model the principle, we appeal to the `weightAdded` predicate defined in the previous section. The *added weight* of a volition subsumes in one value all that comes after it, i.e. all the branches that follow in each considered history. At each time point, the best path across a domain is traced by this number; at each time-point when a choice can be made among alternative volitions, the volition with the highest *added weight* is the only volition which might lead to the best possible outcome. As such, by deeming a volition permissible, the principle establishes that the volition is necessary to reach the best outcome, though it is not usually sufficient. A volition is both necessary and sufficient to reach the best outcome when there are no later volitions which are necessary to reach it. We note that there is no need to calculate the average of the *added weight* of a volition or of its alternatives, since only the greatest one is of interest here. We then postulate that a volition is permissible at a given time if one of its *added weights* at that time is the greatest of any of the *added weights* of its viable alternatives. As before, because of the same history problem, we situate each volition within its simulation, even though the reasoning process is done across simulations. `per(actUti,S,T,I)` is to be read as "at T in S, volition I is the permissible choice because it is one step forward towards the best possible outcome, which may or may not take place in S."

`bestCons(S,T,I,N)`, via the `notBestCons(S,T,I,N)` predicate, determines the highest bound of a partial order inferred from the `weightAdded` values of all the volitions that are possible at a given time point *across all the simulations* which have the same history up to that point. As such, `bestCons(S,T,I,N)` states that, given a volition I available at T in S, the *best possible consequence* of I after T will weigh N.

We then state that a volition I at T in S is impermissible relative to act utilitarianism if there exists

a viable alternative whose *best possible consequence* is greater than its own *best possible consequence*. Any other volition is permissible.

```
notBestCons(S1,T,I,N1):-
    weightAdded(S1,T,I,N1),weightAdded(S2,T,I,N2),sameHistory(T,S1,S2),N1<N2.
bestCons(S2,T,I,N):-
    weightAdded(S1,T,I,N),not notBestCons(S1,T,I,N),occurs(S2,I,T),sameHistory(T,S1,S2).
imp(actUti,S1,T,I1):-
    alterViable(S1,T,I1,I2,S2),bestCons(S1,T,I1,N1),bestCons(S2,T,I2,N2),N1<N2.
per(actUti,S,T,I):-occurs(S,I,T),not imp(actUti,S,T,I),volition(I).
```

Integration We then state that a simulation which contains at least one volition which does not promote act utility is impermissible relative to the principle. This will result in one simulation being deemed permissible, the one exclusively composed of volitions leading to the best outcome (or more than one if multiple simulations contain equivalent best outcomes). Simulations which only contain some permissible volitions contain only sections of plans that are optimal. Note that we could also model this using the equivalent reasoning that the best simulation is the one in which the first volition has the greatest *added weight*.

```
imp(actUti,S):-imp(actUti,S,T,I).
per(actUti,S):-sim(S),not imp(actUti,S).
```

9.2.3 *toR:* Principle of Least Bad Consequence

The principle of least bad consequence states that an action is impermissible if its worst consequence is worse than the worst consequence of any other available action. This principle is particularly relevant to decision-making under uncertainty, where, under the 'bad-luck' assumption that each possible action would yield its worst consequence, the agent may best choose the alternative having the least bad bad consequence [158]. We interpret this principle as a maximising principle rather than a sufficing one, so that we only deem permissible a single volition, the one whose worst consequence is the best among the worst consequences of its viable alternatives (there may also be multiple permissible volitions in the case of a tie). We do not, as in some formulations of the principle, simply prohibit the volition with the worst worst consequence and allow all the others.

This principle is relevant outside of ethics, and is often called maximin or maximum minimorum. It can be used to guide decision-makers looking to maximise the minimum achievable pay-off of investments. Such agents analyse the worst possible outcomes of every choice and select the highest

one, thereby guaranteeing minimal losses, but also potentially missing out on the opportunity to make great profits. In ethical terms, an agent applying this principle typically shows an aversion to doing harm coupled with a low concern for doing good. It is an appropriate guiding rule for a pessimist who aims to achieve the best results while also assuming that the worst will happen or it most likely to happen. It is also appropriate in situations where the possible negative outcomes far outweigh the possible positive outcomes, or where the negative outcomes are sufficiently awful that avoiding them becomes the only concern. Modelling this principle *in foresight* is symmetrical to modelling act utilitarianism *in foresight*.

`worstCons(S,T,I,N)`, via the `notWorstCons(S,T,I,N)` predicate, determines the lowest bound of a partial order inferred from the `weightAdded` values of all the volitions that are possible at a given time point *across all the simulations* which have the same history up to that point. `worstCons(S,T,I,N)` states that, given a volition I available at T in S, the *worst possible consequence* of I after T will weigh N.

We then state that a volition I occurring at T in S is impermissible relative to the principle of least bad consequence if its *worst possible consequence* is worse than the *worse possible consequence* of any viable alternative volition. The volition whose worst possible consequence is the best among the worst possible consequences of all the alternative options will therefore be the permissible one relative to this principle.

```
notWorstCons(S1,T,I,N1):-
    weightAdded(S1,T,I,N1),weightAdded(S2,T,I,N2),sameHistory(T,S1,S2),N1>N2.
worstCons(S2,T,I,N):-
    weightAdded(S1,T,I,N),not notWorstCons(S1,T,I,N),occurs(S2,I,T),sameHistory(T,S1,S2).
imp(leastBad,S1,T,I1):-
    alterViable(S1,T,I1,I2,S2),worstCons(S1,T,I1,N1),worstCons(S2,T,I2,N2),N1<N2.
per(leastBad,S,T,I):-occurs(S,I,T),not imp(leastBad,S,T,I),volition(I).
```

Integration We state that a simulation which contains at least one volition whose worst consequence is worse than any alternative volition is impermissible relative to the principle.

```
imp(leastBad,S):-imp(leastBad,S,T,I).
per(leastBad,S):-sim(S),not imp(leastBad,S).
```

We note that there are two ways in which we might interpret the formula "worst among the worst consequences." We might compare all the possible consequences of a volition and identify the worst

one, or we might compare all the possible *negative* consequences of a volition and among these and pick out the worst one. We chose the first option for the following reasons. First, nothing is specified about this in the typical philosophical formulations of the principle. Second, the "bad-luck" assumption under which the principle is most useful can be relevant to positive outcomes. Consider a student having to choose between two degrees, both of which will have positive outcomes by making her more likely to get certain jobs. Because of a fluctuating market which she cannot predict, she knows that a degree in accounting will at best lead her to be considered for 15 positions and at worst for 5, while a degree in auditing will at best lead her to be considered for 20 jobs and at worst for 2. She is a pessimist, so she applies the principle and chooses accounting. The principle is relevant to this positive outcome. Nevertheless, if for the purpose of modelling a particular scenario the other reading was preferred, this could be done in a simple manner by modifying the definition of `weightCons` so that it only selects event consequences with negative weights.

To a certain extent, this principle is agnostic towards producing good (or better-than-the-worst) outcomes. It says nothing about the consequences of volitions that are not their worst consequences. Consider a situation where the agent has a choice between two volitions which each occur in two different simulations, and whose worst consequence has the same value (say for I1 in s_1 and for I2 s_2). The principle would deem them to be equivalent and permissible. Now consider also that I1 in s_3 produces a very positive outcome, while I2 in s_4 produces a mediocre one. The principle fails to determine that I1 is overall better than I2. An example.

> (*footprint*) A company wants to reduce the carbon footprint of their factory, and has the option of implementing one of two experimental technologies to do so. It is possible that the technologies will not work at all, however if they do work, technology υ will reduce the carbon footprint by 20% and technology ω by 30%.

Hence, in s_1 the company implements υ and nothing changes, in s_2 it implements ω and nothing changes, in s_3 it implements υ and reduces its footprint by 20%, in s_4 it implements ω and reduces its footprint by 30%. As represented in figure 9.3, the principle of least bad consequence, by comparing the worst consequences of the technologies in s_1 and s_2 will see both technologies as equal. To palliate this lacuna, the principle can therefore be usefully coupled with other principles, such as the principle of benefits v. costs.

Finally, we note that this principle could be modelled differently. The `weightAdded` predicate which we use in the model concatenates consequence weights in such a way that the worst *individual* consequence of a volition might be hidden by the presence of other better ones. Take the (*footprint*) example. The domain is the same, except for a precision about ω. As before, ω reduces the company's footprint by 30% in s_4. However, we learn that it reaches this by dramatically increasing

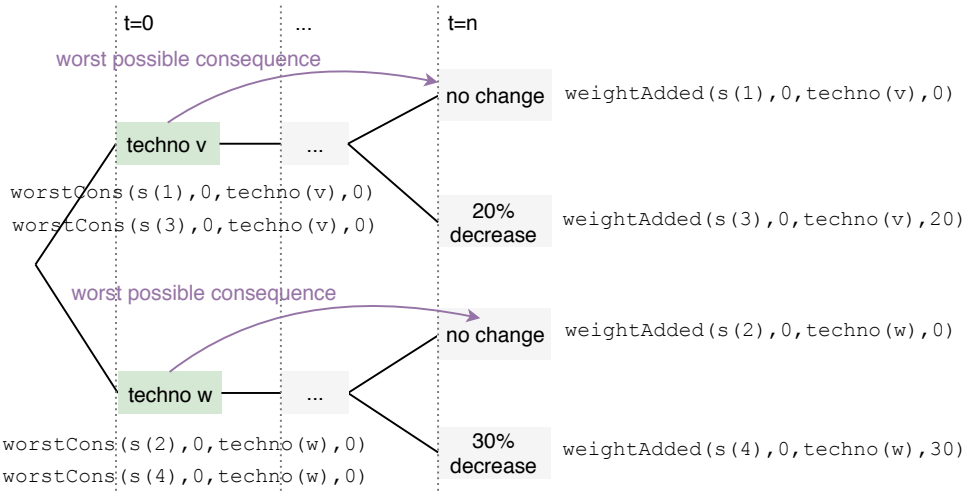

Figure 9.3: Worst possible consequence (*footprint*)

the carbon footprint on one company site and dramatically decreasing it on another location at the same time, resulting in a 30% decrease overall. If we translate this percentage into a straightforward weight value, this corresponds to a `weightAdded` of 30 for ω in s_4. But this number hides two different individual consequences, a carbon increase in location 1 and a carbon decrease in location 2, which correspond to two `weightCons` values of, say, -50 and +80. This is represented in figure 9.4. We could argue that the hostility of the principle faces towards the *worst possible individual consequence* and not towards the *worst possible state of affairs*, so that instead of considering them to be equal, ω should be seen as worse than v, whose worst individual consequence is having no effect is s_1. Typically, this will be preferred in situations where the fact of adding up consequences can be seen as a thorny enterprise. For example, if action A1 kills 10 random people and saves 11 endangered ones, while A2 kills 1, then we might prefer A2 even though it is worse overall, because we believe in the immutable value of each life. Instead, if action A1 makes me loose 10 euros and gain 11, while A2 makes me loose 1, then the former method will be more appropriate. The two methods corresponds to different interpretations of the term "consequence:" is the worst possible consequence the worst event that can occur after I in S or is it the worst set of events that can occur after I in S? The second interpretation (our own) is only interesting in the presence of uncertainty, without it it collapses into act utilitarianism. It is also, we argue, the most fitting here. Focusing on individual consequences rather than overall states of affairs introduces reasoning processes that are closer to deontology than to consequentialism. It demands that we give up concatenating the produced Good in order to focus on a single aspect of the situation. We defer to later works further discussion of this issue.

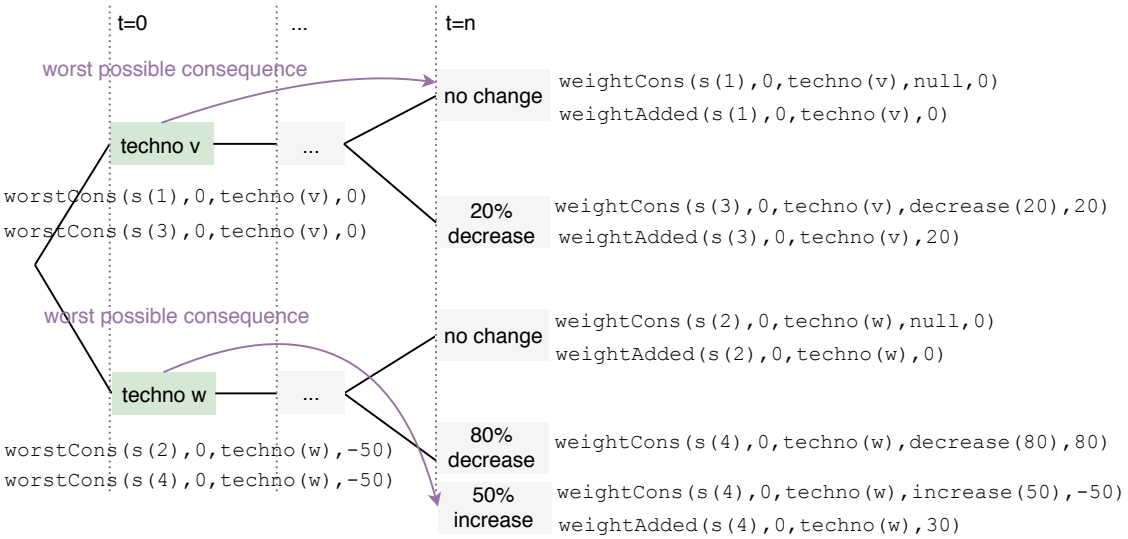

Figure 9.4: Worst possible *individual* consequence (*footprint*)

9.2.4 *toR:* Prohibiting Purely Detrimental Actions

The final volition-based consequentialist principle that we integrate in the framework states that actions with purely detrimental effects are impermissible. This intuitive rule is relevant to most ethical scenarios and is essentially modelled in order to supplement other theories of the Right which may only focus on actions with complex effects. For instance, the doctrine of double effect discussed in section 9.4.3 appraises actions that have both good and bad ramifications, but it makes no explicit reference to actions that are purely detrimental. As such, if modelled alone, the DDE would not deem unacceptable any action that has only bad effects, even though, presumably, proponents of the DDE would not adhere to this judgement. We define the following predicates.

`goodCons(S,I,T)` indicates that volition I occurring at T i S provokes at least one good consequence. This might happen in one of four ways:
- I itself has a positive weight,
- I occurs instead of a viable alternative that has a negative weight,
- I has a counterfactually valid supporting relation with a desirable event,
- I occurs in place of a viable alternative which can have a counterfactually valid supporting relation with an undesirable event.

`badCons(S,I,T)` indicates that volition I occurring at T i S provokes at least one bad consequence. This might happen in one of four ways:
- I itself has a negative weight,

- I occurs instead of a viable alternative that has a positive weight,
- I has a counterfactually valid supporting relation with an undesirable event,
- I occurs in place of a viable alternative which can have a counterfactually valid supporting relation with a desirable event.

Because of the reflexive quality of *causing*, which translates into reflexivity for causal properties, we do not need to explicitly model the first two cases for producing good and bad consequences, but can subsume them in the last two cases, as follows.

```
goodCons(S,T,I):-valid(S,R,I,T,E),weight(E,N),N>0,allSupRel(R),volition(I).
goodCons(S1,T,I1):-
    alterViable(S1,T,I1,I2,S2),valid(S2,R,I2,T,E),weight(E,N),N<0,allSupRel(R).
badCons(S,T,I):-valid(S,R,I,T,E),weight(E,N),N<0,allSupRel(R),volition(I).
badCons(S1,T,I1):-
    alterViable(S1,T,I1,I2,S2),valid(S2,R,I2,T,E),weight(E,N),N>0,allSupRel(R).
```

All types of relationships are taken into account, be they direct or transitive. We only appeal to supporting relations and not to opposing ones for the reasons discussed previously relative to the other consequentialist principles. The specification of counterfactual validity is important here to remove certain false positives, such as the finding that a volition has a good consequence when in fact it does not. Take an action which *causes* a bad outcome (and has no good consequences), such as switching a train from tracks A where there is nothing onto tracks B where there is a dog. This is a purely bad action. However, if we were to add that this action enables another action whose consequences would be bad if it were performed, but that the agent does not perform it, the existence of this second action will mean that the first action is no longer purely bad: it *helps to exclude* some bad outcomes. This would be the case if after flipping the first switch and letting the train run into the dog, the agent then could flip a second switch onto tracks C where there is a cow. Surely the first switch action is not morally better than before just because there are added risks; if anything, it is worse. Checking for counterfactual validity ensures these cases are not taken into account: if the first switch action *helps to exclude* crashing into the cow, the relation is counterfactually invalid since, had the switch not occurred, that crash would not have occurred anyway.

As noted in the *causal model*, omissions that occur in the presence of more than one viable alternative action cannot be tested for counterfactual validity since there is no way of knowing what the opposite of the omission is among the multiple possibilities. For this reason we add omission-specific rules which do not make a demand for counterfactual validity but instead ascertain the less demanding fact that the relation is not extrinsically necessary.

```
goodCons(S,T,O):-notNecessary(S,R,exc,O,T,E),weight(E,N),N>0,allSupRel(R),omission(O).
goodCons(S1,T,I):-
    alterViable(S1,T,I,O,S2),notNecessary(S2,R,exc,O,T,E),weight(E,N),N<0,allSupRel(R),
    omission(O).
badCons(S,T,O):-notNecessary(S,R,exc,O,T,E),weight(E,N),N<0,allSupRel(R),omission(O).
badCons(S1,T,I):-
    alterViable(S1,T,I,O,S2),notNecessary(S2,R,exc,O,T,E),weight(E,N),N>0,allSupRel(R),
    omission(O).
```

We then state that an action is impermissible if it only has bad consequences, and that any action that has not been shown to be impermissible is by default permissible relative to this principle. Like the previous three, this principle is modelled in foresight, so that good and bad consequences are investigated in all branches of the tree that emanate from a volition choice. Therefore, even if a volition is purely bad in simulation, it will still be permissible if there also exists a simulation in which it is not.

```
imp(pureBad,S3,T,I):-
    badCons(S1,T,I),not goodCons(S2,T,I),sameHistory(T,S1,S2),sameHistory(T,S2,S3).
per(pureBad,S,T,I):-occurs(S,I,T),not imp(pureBad,S,T,I),volition(I).
```

Integration We state that a simulation which contains at least one volition that is purely bad and no volitions that are not purely bad is impermissible relative to this principle.

```
imp(pureBad,S):-imp(pureBad,S,T,I),not per(pureBad,S,T,I).
per(pureBad,S):-sim(S),not imp(pureBad,S).
```

To conclude the discussion of these four consequentialist principles, we summarise in table 9.1 the `weight` predicates that enable us to model them.

9.2.5 Consequentialist Principles and Decision Theory

The challenge of computationally applying ethical principles to entire plans of actions rather than to isolated volitions has made salient some compelling comparisons. It has made visible the resemblances that some consequentialist principles share with well-know probabilistic rules and frameworks employed in game theory and decision theory under uncertainty. As discussed in their individual sections, the principle of benefits v. costs, act utilitarianism and the principle of least bad consequence, when modelled *in foresight*, can be apprehended as the principle of insufficient reason (also called principle of indifference), maximax and maximin respectively.

Table 9.1: Summary of weight predicates

Predicate	Description
`weight(E,N)`	E weighs N in virtue of a theory of the Good
`weightCons(S,T,I,E,N)`	I at T in S has as a supported consequence E which weighs N
`weightAdded(S,T,I,N)`	I at T in S has an *added weight* of N, which is the sum of its supported consequences
`weightPossible(S1,T,I,N,S2)`	I at T might turn out to have an *added weight* of N
`weightAv(S,T,I,N)`	I at T has *average added weight* of N
`alterAv(S,T,I,N)`	I at T is to be compared to the *average added weight* N, of its viable alternatives
`weightCompared(S,T,I,N)`	I at T has a *compared weight* of N
`bestCons(S,T,I,N)`	I at T can at best assume an *added weight* of N, given possible future volitions
`worstCons(S,T,I,N)`	I at T can at worst assume an *added weight* of N, given possible future volitions
`goodCons(S,T,I)`	I at T has at least one *possible consequence* with a *weight*>0
`badCons(S,T,I)`	I at T has at least one *possible consequence* with a *weight*<0

The Principle of Insufficient Reason This principle represents chance. It states that if there is no reason to distinguish different possibilities beyond inconsequential factors such as their names, then, given n possibilities, each one should be assigned a probability of $1/n$. Typically, this occurs when knowledge of a system is too poor to predict its results. As John Arbuthnot says in *Of the Laws of Chance* [9],

> "It is impossible for a Die, with such determin'd force and direction, not to fall on such determin'd side, only I don't know the force and direction which makes it fall on such determin'd side, and therefore I call it Chance, which is nothing but the want of art..."

In Bayesian statistical inference, this principle is the simplest diffuse prior, a probability distribution which expresses vague or general beliefs about a variable [133]. In the present framework, we apply this principle to future volition choices (we do not apply it to automatic events themselves, for these are deterministic). Taking the subjective view of an agent choosing volition I at T, the principle serves us to evaluate the probability that a set of events will occur by determining the probability that affector volitions will be chosen by agents after T. In this case, the probability of occurrence for n available volitions at T is $1/n$ at every T. Relative to ethical decision-making, this rule will typically be useful for an agent with no knowledge of the future and no reasons to believe that some later volitions have a greater chance of occurring than others.

Maximax The maximax strategy seeks out the greatest possible benefit [180]. In game theory, it corresponds to adventurous decisions-makers who aim for high payoffs and who are in a position to

withstand loss without critical inconvenience. Applied to the ethical evaluation of a volition within a plan, this strategy will be a good one for an agent who trusts that the volitions that will be chosen at later times will promote the Good. This will be the case if the agent knows that they themselves will perform the choice, or if they have a degree of trust in the agent who will. We note one possible distinction that can be made between the maximax rule and act utilitarianism modelled *in foresight*. Similarly to the discussion on the principle of least bad consequence, maximax can be applied in one of two ways. Either the agent favours the best possible outcome overall or it favours the best possible individual consequence (so that the less good effects of a volition are ignored). The first method is the one implemented in act utilitarianism using `weightAdded`, while the second method would consist in creating a predicate which tracks the best possible single event emanating from a volition choice. Pertaining to applications of game and decision theory, both are potentially useful. However, it is contradictory to apply act utilitarianism only to the best consequence. If A1 saves 1 person but kills 10 while A2 kills just 2, this method would incite the agent to prefer A1 over A2 in virtue of the one saved, which evidently does not promote the Good.[2]

Maximin The maximin strategy demands that the decision-maker only consider the minimum payoff of each alternative and choose among them the best one [243]. It corresponds to cautious agents who seek a guaranteed known minimum payoff. It can be particularly useful in cases where poor outcomes are the most likely or in order to avoid some very unfavourable worst possible outcomes. But it is also relevant to decisions that do not involve uncertainty. In policy making, for instance, a social policy might be justified on the basis that it advantages the least advantaged members of an unequal society, thereby maximising the *minimal* standards of life [242]. This is an idea which pervades Rawls' view of moral philosophy [197]. In terms of the present model, it will typically appeal to an agent who believes that later volitions will be chosen by ill-willed or untrustworthy agents.

Though relatively straightforward, the similarities between these three ethical principles and decision rules were not obvious at first, nor are they, to our knowledge, much discussed in extant philosophical works. The analytic process of transforming natural language philosophical discourses into explicit logical rules has shone light on it. It has allowed us to appraise ethical doctrine through a new lens, but also show that the principle of benefits v. costs, act utilitarianism and the principle of least bad consequence have more in common than might be immediately conveyed by philosophers. Because they can be described in game or decision-theoretic terms, and formalised using similar tools, these three principles are also linked in a way that other ethical principles are not. We

[2]This points to an asymmetry between act utilitarianism and the principle of benefits v. costs. The former is only meaningful when applied to states of affairs, while the latter can also be meaningful when applied to individual consequences.

summarise this in table 9.2. Moreover, though they do not correspond to explicit ethical principles put forwards by philosophers, other related decision rules exist and could be modelled and applied to ethical dilemmas. They include, among others, Hurwicz's optimism/pessimism criterion [122], which seeks to find a varying middle ground between maximax and maximin by using a coefficient of optimism, or Savage's minimax-regret, which seeks to minimise regret by making the agent consider in advance the regret they would feel, given every choice for every possible final outcome [212].

Table 9.2: Consequentialist principles and decision theory

Principle	Decision Rule	Agent Outlook	Direction
benCosts	Insufficient Reason	neutral	highest possible *compared weight*
actUti	Maximax	optimistic	highest possible *added weight*
leastBad	Maximin	pessimistic	highest possible *worst added weight*

We finally note that we can also visualise these principles as decision trees (branching time or other). A decision tree is a graphical representation of the logical and temporal sequence of events implicated in a decision process [226]. Typically arranged from left to right, it is composed of branches and nodes, within which the first ("decision") node represents the decision being analysed, chance nodes stand for events that are somewhat determined by chance, and terminal nodes for final outcomes. As show in figure 9.5, these are traditionally represented by squares, circles and triangles respectively. Positive and negative payoffs are then associated with the decisions made at the choice nodes and at the terminal nodes. The optimal choice at the decision node can then be inferred or approached by appealing to the values of these payoffs and their probabilities of occurrence contained in the chance nodes. To solve the tree in such a way, one works backward from right to left by *folding back* the tree. This is essentially what we do when working from the `weight` predicate towards an evaluation of a volition as permissible or impermissible, through such intermediary predicates as `weightAdded`, `weightCompared` or `worstCons`. Each decision node corresponds to a considered volition, each chance node to a later volition choice, each terminal node to the state of the world at the end of a simulation, and each payoff to the `weight` of events that occur in a simulation (given by a theory of the Good). Because we might be interested choosing between volitions at any time in a simulation, these can be considered decision nodes at any time, that is, at any stage of the tree.

We illustrate this comparison using the (*collision*) example, as recalled in figure 9.6. Iris assigns weights to different events according to her theory of the Good as discussed in the proof of concept #3, of which we give a superficial summary. She considers that death weighs –100 and reaching a full recovery weighs +100. In addition, she considers that the different ways of dying or of reaching some form of recovery are not equal. Creating an accident in itself weighs –60 because it might injure someone else, a slow decline to deaths weighs –10 because it degrades one's dignity, going

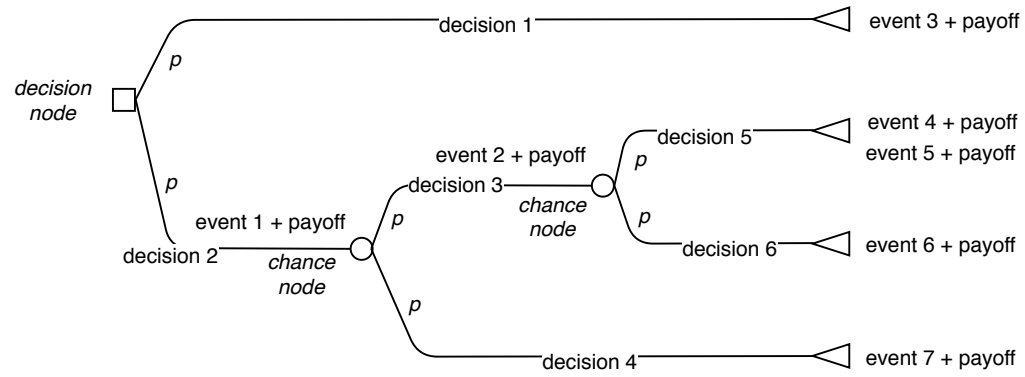

Figure 9.5: A generic decision tree

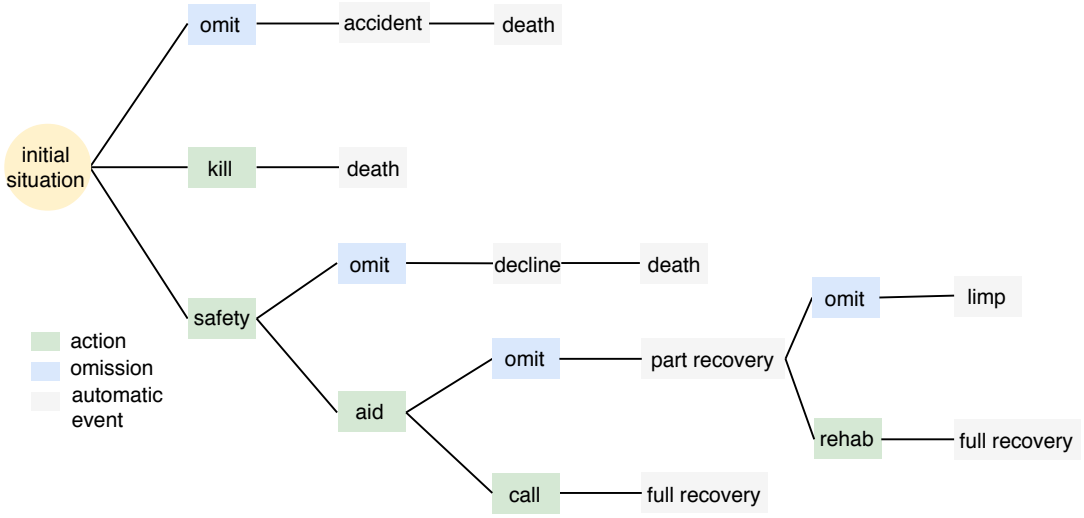

Figure 9.6: An ethical decision-making domain illustrated (*collision*)

through a partial recovery weighs –10 because it is very painful, and ending up with a limp weighs +56 since it is not as good as a full recovery. Finally, she believes that certain actions display virtues of character, and she gives each manifestation of a virtue a value of +10. An agent who gives first aid to a victim displays *knowledge*, and an injured person who goes to rehab displays *determination*. Figure 9.7 shows a translation of this into a decision tree. The probabilistic values given for each decision correspond to her appraisal of the situation through the principle of benefits v. costs, which considers that all volitions are equally likely to be chosen among available choices. What differs here from a typical decision tree is the simplistic nature of the probabilities given to the chance node choices, which are either equally distributed among alternatives (benCosts), assign

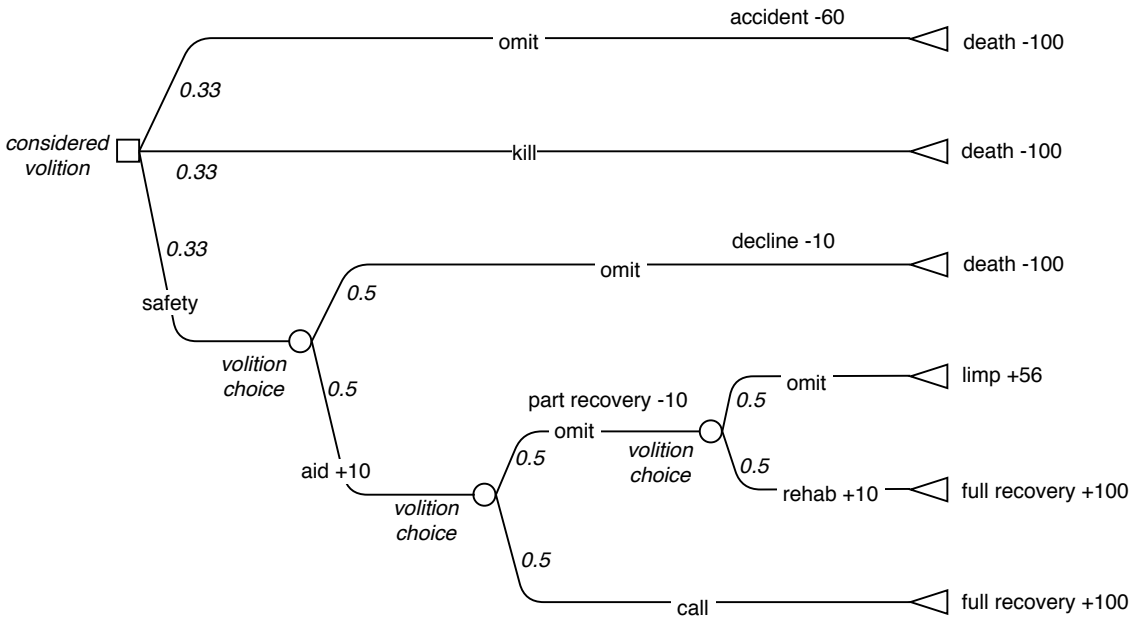

Figure 9.7: An ethical decision-making domain as a decision tree (*collision*)

100% to the best choices and 0% to the others (actUti), or assign 100% to the worst choices and 0% to the others (leastBad). Though it is beyond the scope of the present work, we could imagine modelling scenarios in which the probability distribution is much more complex, for instance by taking into account agent intentions, refined agent capabilities, inter-agent affinities, etc. This could then also motivate us to apply more advanced machine learning techniques to the tree analysis and enhance the decision-making process.

9.2.6 Proof of Concept #4 (*collision*)

We now turn to the translation of this case and decision tree into answer set form. For questions of readability and because there are no problematic cases of diverging histories, we show results of volition permissibility for each time point, but without keeping the simulation variable. For example, `per(actUti,s(2),0,act(iris,safety(driver)))` becomes `per(actUti,0,act(iris,safety(driver)))`, which is to be read as "according to act utilitarianism, Iris's safety action is permissible at t=0 across all simulations."

```
% --------- Principle of benefits v. costs
% Permissible
per(benefitsCosts,0,act(iris,safety(driver))).
```

```
per(benefitsCosts,1,act(iris,aid(driver))).
per(benefitsCosts,2,act(iris,call)).
per(benefitsCosts,4,act(driver,rehab)).
% Impermissible
imp(benefitsCosts,0,act(iris,kill(driver))).
imp(benefitsCosts,0,omit(iris,0)).
imp(benefitsCosts,1,omit(iris,1)).
imp(benefitsCosts,2,omit(iris,2)).
imp(benefitsCosts,4,omit(driver,4)).
% --------- Act Utilitarianism
% Permissible
per(actUti,0,act(iris,safety(driver))).
per(actUti,1,act(iris,aid(driver))).
per(actUti,2,act(iris,call)).
per(actUti,2,omit(iris,2)).
per(actUti,4,act(driver,rehab)).
% Impermissible
imp(actUti,0,act(iris,kill(driver))).
imp(actUti,0,omit(iris,0)).
imp(actUti,1,omit(iris,1)).
imp(actUti,4,omit(driver,4)).
% --------- Principle of least bad consequence
% Permissible
per(leastBad,0,act(iris,kill(driver))).
per(leastBad,1,act(iris,aid(driver))).
per(leastBad,2,act(iris,call)).
per(leastBad,4,act(driver,rehab)).
% Impermissible
imp(leastBad,0,act(iris,safety(driver))).
imp(leastBad,0,omit(iris,0)).
imp(leastBad,1,omit(iris,1)).
imp(leastBad,2,omit(iris,2)).
imp(leastBad,4,omit(driver,4)).
% --------- Prohibiting purely bad volitions
% Permissible
per(pureBad,0,omit(iris,0)).
per(pureBad,0,act(iris,kill(driver))).
per(pureBad,0,act(iris,safety(driver))).
```

```
per(pureBad,1,omit(iris,1)).
per(pureBad,1,act(iris,aid(driver))).
per(pureBad,2,act(iris,call)).
per(pureBad,2,omit(iris,2)).
per(pureBad,4,omit(driver,4)).
per(pureBad,4,act(driver,rehab)).
```

We can visualise this information in the following table.

Table 9.3: Proof of Concept: ToR (*collision*)

Time	Event	benCosts	actUti	leastBad	pureBad
0	`act(iris,safety(driver))`	✓	✓	•	✓
0	`act(iris,kill(driver))`	•	•	✓	✓
0	`omit(iris,0)`	•	•	•	✓
1	`act(iris,aid(driver))`	✓	✓	✓	✓
1	`omit(iris,1)`	•	•	•	✓
2	`act(iris,call)`	✓	✓	✓	✓
2	`omit(iris,2)`	•	✓	•	✓
4	`act(driver,rehab)`	✓	✓	✓	✓
4	`omit(driver,4)`	•	•	•	✓

✓for permissibility and • for impermissibility

We can note a number of dynamics visible on the table, such as the fact that no volition is purely bad here or that at t=2, both volitions are permissible according to act utilitarianism because they can both lead to the best possible outcome (full recovery). A particularly interesting dynamic can be found in the decision that Iris must make at t=0, when she has to choose between killing the driver, bringing him to safety, or doing nothing. Relative to the principle of benefits v. costs, the permissible action is bringing the driver to safety because it has the highest *compared weight* among possible alternatives. Again we take out the simulation variable.

```
weightCompared(0,omit(iris,0),-1732).
weightCompared(0,act(iris,kill(driver)),-1012).
weightCompared(0,act(iris,safety(driver)),1715).
```

The same is true for act utilitarianism, because it has best *possible consequence*, in other words, the best case scenario necessarily goes through this action.

```
bestCons(0,omit(iris,0),-160).
```

```
bestCons(0,act(iris,kill(driver)),-100).
bestCons(0,act(iris,safety(driver)),110).
```

However, relative to the principle of least bad consequence, the permissible action is killing the driver. Indeed, bringing him to safety carries the risk that he will die a slow, agonising death (–110), rather than a rapid one (–100). She might see fit to kill him if she thought, for instance, that she would not have the capacity to perform first aid. Just as one might put a wounded animal out of its misery, so would Iris bestow the gift of a painless death upon the driver were she to apply this principle.

```
worstCons(0,omit(iris,0),-160).
worstCons(0,act(iris,kill(driver)),-100).
worstCons(0,act(iris,safety(driver)),-110).
```

In this section, we have investigated four principles that base their consequentialist calculation on volitions, then have extended them to simulations as a whole. We now turn to a principle which appeals not to immediate individual choices, but to general rules that span across simulations.

9.3 Consequentialist Ethics Based on Rules

9.3.1 *toR:* Rule Utilitarianism

> *"Each act, in the moral life, falls under a rule; and we are to judge the rightness or wrongness of the act, not by its consequences, but by the consequences of its universalisation - that is, by the consequences of the adoption of the rule under which this act falls" J. Hospers, 1975 [120]*

According to rule utilitarianism, the moral assessment of an action consists in a two-step procedure [230]. The first step consists in the appraisal of moral rules on the basis of the principle of utility: one must determine whether a moral rule (or set of moral rules), will lead to the best overall consequences, assuming all, or at least most agents follow it. In everyday life, likely such rules may include "Do not steal," or "Keep your promises." The second step consists in the appraisal of particular actions in the light of what has been justified during the first step. One can perform a concrete action in a specified situation only if the action is sanctioned by a rule that was determined to uphold the principle of utility, whether or not the action itself adheres to the principle of utility. For example, if "Do not steal" has been adopted, then stealing will always be impermissible, even in cases where the particular instance of stealing would produce the greatest utility (say because it will feed a starving child). Unlike with act utilitarianism, the issue is not which *action* produces

the greatest utility, but which *moral rule* does. By extension, the principle also investigates the general social practice that can result from widespread compliance to such a rule.

Rule utilitarianism was first formulated by Roy Forbes Harrod in 1936 [104], with the diverging terms of "act utilitarianism" and "rule utilitarianism" later introduced by Richard Brandt in 1959 [34]. It purports to solve some of the issues that act utilitarianism faces, such as the fact that it can be viewed as unreasonably demanding. For instance, the principle suggests that, given the state of the world and existing inequality among people, any person living an average life in a more economically developed country would have the duty to give away almost all of their income to much poorer people, since the benefit of this money to them (e.g. comfort) would be far lesser than the benefit to the recipient (e.g. survival). Arguably, it presents as duties what would usually be considered to be over-abounding self-sacrifice. Because rule utilitarianism assumes that everyone, or at least most people, adhere to a given accepted rule, it demands less of each individual. If everyone accepted the rule "unless you are extremely poor, give away some of your money until extreme poverty is eradicated," then each agent would have to part with a smaller portion of their income [115]. Act utilitarianism is also problematic in that it may also licence distinctly reprehensible behaviour. For example, if I were to consider that my friend's computer would bring me greater happiness than it brings them, then I would be right to steal it in so far as it will bring more overall satisfaction [106]. Another criticism is that it can be too difficult to apply in practice. No person will know all the possible consequences of their actions, nor will they typically have enough time to compute all the likely consequences of a particular choice before making it, nor will they always be rational enough to reach valid conclusions[3]. Humans have for instance shown to display a bias which tends to minimise the harm done to others by acts that benefit them [115]. By adopting a manageable set of rules before considering particular actions, the agent can bypass this issue. Rule utilitarianism also addresses some problems that arise in conventional theories of rational behaviour [107]. For example, most accounts of the sort lead to the paradoxical notion that, given any large pool of voters, voting is an irrational activity in so far as no individual vote is ever likely to influence the result of the election. Rule utilitarianism succeeds in providing a rational principle for doing so, by arguing that a rational commitment underpins the action.

There exist a number of versions of rule utilitarianism. We here focus on modelling the most widespread version, which is concerned with the acceptance of the rule by *all* agents, rather than most. We define the following predicates.

`universal(S,R)`, through the `notUniversal(S,R)` predicate, identifies the simulations in which a particular rule has been universalised. These are simulations in which every instance of a rule occurs any time it can.

[3]Presumably, this criticism also applies to artificial autonomous agents, though perhaps to a lesser extent.

`simEvents(S,E,T,N)` identifies all the events that occur in a particular simulation and traces their weights N.

`simEndstate(S,N)` sums all the weighted events that occur in any given simulation. We call this value the simulation's *end-state weight*.

`ruleEndstate(S,R,N)` traces, for a given rule R, the *end-state weight* if the simulations in which R has been universalised.

`ruleSum(R,N)` sums, for a given rule R, the *end-state weights* of all the simulations in which R has been universalised. There may be more than one such simulation for each given rule, since, as long as the rule has been universalised, any possible configuration of other volitions on top of it will produce a separate simulation.

`ruleCount(R,N)` counts, for a given rule R, the number of *end-state weights* of all the simulations in which R has been universalised.

`ruleWeight(R,N)` gives, for a given rule R, the average *end-state weight* of all the simulations in which R has been universalised. We call this value the *rule weight*.

We note here that just like rule utilitarianism focuses on general outcomes but not on individual volitions, so does the formalisation of the principle refrain from appealing to volitions and reasons directly over simulations. We then state that a volition I occurring at T in S is impermissible relative to rule utilitarianism if it is an instance of a rule whose *rule weight* is inferior to the *rule weight* of a rule of which one of its viable alternative volitions is an instance. Any other volition is permissible relative to the principle. Formulating the principle in such a way means that a partial order is determined between rules and that an agent will always be able to chose among available options the *best* one. This is not to say the agent chooses a *good* volition. If the choice is to be made between exclusively negative *rule weights*, then utility will not be maximised, instead it will be minimally minimised. We also consider that if it is not specified whether a volition is the instance of a rule, then this volition cannot be appraised by this principle as either permissible or impermissible. Note that rules and rule instances are to be defined as `rule(R)` and `instance(I,R)` in the modality specification module.

```
notUniversal(S,R):-complete(S,I,T),not occurs(S,I,T),instance(I,R).
universal(S,R):-not notUniversal(S,R),sim(S),rule(R).
simEvents(S,E,T,N):-occurs(S,E,T),weight(E,N).
simEndstate(S,N1):-N1=#sum[simEvents(S,E,T,N2)=N2],sim(S).
ruleEndstate(S,R,N):-universal(S,R),simEndstate(S,N).
ruleSum(R,N1):-N1=#sum[ruleEndstate(S,R,N2)=N2],rule(R).
ruleCount(R,N1):-N1=#count{ruleEndstate(S,R,N2)},rule(R).
ruleWeight(R,N1/N2):-ruleSum(R,N1),ruleCount(R,N2),N2!=0.
```

```
ruleWeight(R,0):-ruleCount(R,0).
imp(ruleUti,S1,T,I1):-
    occurs(S1,I1,T),alterViable(S1,T,I1,I2,S2),instance(I1,R1),ruleWeight(R1,N1),
    instance(I2,R2),ruleWeight(R2,N2),N1<N2.
per(ruleUti,S,T,I):-occurs(S,I,T),instance(I,R),not imp(ruleUti,S,T,I),volition(I).
```

Integration We state that a simulation which contains at least one volition which is impermissible relative to the principle is itself impermissible. In other words, the only simulations that are permissible as a whole are the ones in which each chosen volition belongs to the most utility maximising rule among available choices. This means that the most utility maximising rule will be universalised, but that other good rules might not be, since they might be overtaken by better ones where possible.

```
imp(ruleUti,S):-imp(ruleUti,S,T,I).
per(ruleUti,S):-sim(S),not imp(ruleUti,S).
```

9.3.2 Proof of Concept #5 (*venture*)

Not all scenarios can be relevantly looked upon by rule utilitarianism. Typically, the principle will only be workable if the same action or type of action can be performed by multiple different agents. Likewise, it is not the case that all volitions can be compellingly extended to a rule. Turning left at every stoplight is not, *a priori*, an interesting rule. The role of omissions is also complex. Though omissions can occur in situations which have nothing in common with one another, it might be interesting to reason over a general rule of omitting to act, since it will bring to light the dynamics that govern an environment in the absence of agents. More restrained rules of omitting to act *in certain types of situations* could also yield telling results. For example, the political inertia of individual citizens acquires a different meaning under democratic or dictatorial regimes. For all these reasons, we provide a specific example and proof of concept tailored to the applicability of rule utilitarianism. Consider the following case adapted from [107].

> (*venture*) 5 benefactors have to decide the fate of a new desirable social venture which promotes education. All of them favour it. Yet it will go ahead only if all benefactors actually decide to pitch in a lump sum. Therefore, if one or more of them chooses not to invest, the others' money is lost. But they also have the possibility of directly giving their money to an existing confirmed education charity, which will directly and certainly use it. The benefactors cannot communicate and cannot find out what the others have done or will choose to do. The overall positive impact of the social venture is greater than the impact that the charity will make if all the money goes to it.

In order to model the case, we also specify the following elements. To avoid generating a huge amount of unnecessary simulations, we consider that each agent has a slot for acting, so that each benefactor can only make her decision of investing in the venture, donating to charity or doing nothing once the previous benefactors already has. To translate the fact that the venture is more beneficial than donations, we note in the theory of the Good that, while both participate in promoting the right to education for a group of students (considered as one entity), the venture promotes it by 80% while each individual donation promotes it by 10%. We then consider that each action corresponds to a rule of the same name, and that all omissions correspond to a passive "let it be" rule.

```
% -------------- Scenario Generation
sim(simName).
0{performs(simName,A,T):action(A)}1:-time(T).
% -------------- Domains
number(-100..100).
time(0..6).
agent(g(1);g(2);g(3);g(4);g(5);students).
positiveFluent(invested(G);can(G,invest);can(G,donate)):-agent(G).
nonInertial(can(G,invest);can(G,donate)):-agent(G).
% -------------- Initial Situation
initially(can(g(1),invest)):-agent(G).
initially(can(g(1),donate)):-agent(G).
% -------------- Event Specification
% inTurn
effect(act(g(N),X1),can(g(N+1),X2)):-agent(g(N)),actName(X1;X2).
effect(omit(g(N),T),can(g(N+1),X)):-agent(g(N)),actName(X),time(T).
effect(act(g(N),X1),neg(can(g(N),X2))):-agent(g(N)),actName(X1;X2).
effect(omit(g(N),T),neg(can(g(N),X))):-agent(g(N)),actName(X),time(T).
% Act: invest
capable(g(1..5),invest).
prec(can(G,invest),act(G,invest)):-agent(G).
effect(act(G,invest),invested(G)):-agent(G).
% Act: donate
capable(g(1..5),donate).
prec(can(G,donate),act(G,donate)):-agent(G).
% Auto: launch
auto(launch).
prec(invested(G),launch):-capable(G,invest).
```

```
effect(launch,neg(invested(G))):-capable(G,invest).
% --------- Target Specification
targetNumber(students,1).
targetWeight(students,1).
% --------- Modality Specification
% rights
right(education).
modalityWeight(education,100).
% rules
rule(R):-instance(I,R).
instance(O,letItBe):-omission(O).
instance(act(G,X),X):-capable(G,X).
% --------- Ethical Event Specification
impact(launch,students,education,80).
impact(act(G,donate),students,education,10):-agent(G).
```

As a note, the lack of a G variable attached to `launch` in `prec(invested(G),launch):-agent(G)` captures the demand that all agents must invest in order for it to occur. A look at the answer set shows that there are 243 different simulations corresponding to a possible configuration of each benefactor choosing one of the three possible volitions at their allotted time. It also tells us that each rule has been universalised in a single simulation – since no extra volition choices are ever available to the agents – and that their *rule weights* are 80, 50 and 0 for *investing*, *donating* and *letting it be*, respectively. Accordingly, under rule utilitarianism, every instance of donating and omitting to act is impermissible and every instance of investing is permissible.

```
universal(s(62),invest).
universal(s(94),donate).
universal(s(219),letItBe).
ruleWeight(invest,80).
ruleWeight(donate,50).
ruleWeight(letItBe,0).
occurs(s(62),act(g(1),invest),0).
occurs(s(62),act(g(2),invest),1).
occurs(s(62),act(g(3),invest),2).
occurs(s(62),act(g(4),invest),3).
occurs(s(62),act(g(5),invest),4).
occurs(s(62),launch,5).
imp(ruleUti,s(37),0,act(g(1),donate)).
```

```
imp(ruleUti,s(19),1,act(g(2),donate)).
imp(ruleUti,s(1),0,omit(g(1),0)).
imp(ruleUti,s(243),1,omit(g(2),1)).
per(ruleUti,s(243),0,act(g(1),invest)).
per(ruleUti,s(198),1,act(g(2),invest)).
per(ruleUti,s(62)). (etc.)
```

Comparison with Other Principles

It is interesting to compare these results with the claims made by other theories. In particular, act utilitarianism prescribes a different but appealing set of acceptable volitions. It states that donating to charity is permissible only if launching the venture is no longer possible, i.e. if another agent has already been given the chance to but has failed to invest. Reversely, investing is permissible only if launching the venture is still possible, in which case it is always permissible. Omitting to act is never permissible, because donating is always better than doing nothing. As represented in table 9.4, simulation s_{157} gives a good illustration of this dynamic. (Nevertheless, we note that act utilitarianism will only be of use to these agents if they have a way of knowing the choices that others made before them.)

Table 9.4: Proof of Concept: ToR (*venture* s_{157})

Time	Event	ruleUti	actUti	benCosts	leastBad	rule weight	co. weight
0	`act(g(1),invest)`	✓	✓	•	•	+80	–40
1	`omit(g(2),1))`	•	•	•	•	0	–64
2	`act(g(3),donate))`	•	✓	✓	✓	+50	+100
3	`act(g(4),invest))`	✓	•	•	•	+80	–50
4	`act(g(5),donate))`	•	✓	✓	✓	+50	+100

✓for permissibility and • for impermissibility

Next, given the low probability of launching the venture among all possible outcomes, the principle of benefits v. costs yields another set of permissible volitions, which here encourages donating. This first agent to act is never permitted to invest relative to this principle, for this action only ever has a positive *compared weight* once at least the two previous agents have already invested. In other words, it only becomes permissible when launching the venture becomes more likely than unlikely. As such, the first two agents are only permitted to donate, and it is only if they chose this impermissible action that the next agents are then permitted to invest. In addition, once launching the venture is no longer possible because at least one benefactor has failed to invest, donating to charity always has a *compared weight* of 100, since alternatives lead to no benefit at all. Finally, the principle of least bad consequence gives the same results as the previous principle but for different

reasons. It considers that donating is the only volition which is permissible in all instances, except for one instance of investing in one simulation. This instance is the one performed by the fifth agent after the previous four have done so too; it consists in the only case in which investing carries no risk of failure.

```
% s(157)
occurs(s(157),act(g(1),invest),0).
occurs(s(157),omit(g(2),1),1).
occurs(s(157),act(g(3),donate),2).
occurs(s(157),act(g(4),invest),3).
occurs(s(157),act(g(5),donate),4).
% --------- Rule utilitarianism
per(ruleUti,s(157),0,act(g(1),invest)).
imp(ruleUti,s(157),1,omit(g(2),1)).
imp(ruleUti,s(157),2,act(g(3),donate)).
per(ruleUti,s(157),3,act(g(4),invest)).
imp(ruleUti,s(157),4,act(g(5),donate)).
% --------- Act utilitarianism
per(actUti,s(157),0,act(g(1),invest)).
imp(actUti,s(157),1,omit(g(2),1)).
per(actUti,s(157),2,act(g(3),donate)).
imp(actUti,s(157),3,act(g(4),invest)).
per(actUti,s(157),4,act(g(5),donate)).
% --------- Principle of benefits v. costs
imp(benefitsCosts,s(157),0,act(g(1),invest)).
imp(benefitsCosts,s(157),1,omit(g(2),1)).
per(benefitsCosts,s(157),2,act(g(3),donate)).
imp(benefitsCosts,s(157),3,act(g(4),invest)).
per(benefitsCosts,s(157),4,act(g(5),donate)).
% --------- Principle of least bad consequence
imp(leastBad,s(157),0,act(g(1),invest)).
imp(leastBad,s(157),1,omit(g(2),1)).
per(leastBad,s(157),2,act(g(3),donate)).
imp(leastBad,s(157),3,act(g(4),invest)).
per(leastBad,s(157),4,act(g(5),donate)).
% --------- Compared weight
weightCompared(s(157),0,act(g(1),invest),-40).
weightCompared(s(157),1,omit(g(2),1),-64).
```

```
weightCompared(s(157),2,act(g(3),donate),100).
weightCompared(s(157),3,act(g(4),invest),-50).
weightCompared(s(157),4,act(g(5),donate),100).
```

Relative to simulations as a whole, act utilitarianism gives the same response as rule utilitarianism by considering that only the simulation in which all agents invest is permissible, since it abides by the optimistic view that all agents will make the best possible choice. The principle of benefits v.costs divides simulations into two large groups (permissible or impermissible) based on the risks taken by each one of the agents in changing contexts. The principle of least bad consequence instead deems that the permissible simulation is the one in which all benefactors donate, since this brings much Good while avoiding all risk of failing to do so.

The present module corresponds to one possible version of rule utilitarianism, but a number of variants exist. Though we do not model them here, we quickly survey a number of them and note that they present promising avenues for future works. Some philosophers argue that the most plausible version of rule utilitarianism is the one which limits the universalisation of rules to less than 100% of a given population, so that the inescapable existence of non-compliers can be efficiently taken into account. If 100% compliance is never in fact possible, or even just occasionally challenged, then there is arguably no point in basing our moral reasoning on such a case. Consider the following example from [115].

> You and I are walking to the airport when we see two small children drowning in a lake. You and I could easily save the children, at no risk to ourselves. The two children are positioned in the lake in such a way that you and I could each save one and still get to our flights. But if one of us saves both children, he will miss his flight. Suppose you save one child, but I do nothing.

Surely, you should now save the other. Basing the universalisation of rules on full compliance would mean that rule utilitarianism could not make this demand, since such a case would not be considered. Saving the second child would not be required, since it is assumed that everyone does their bit but needn't go beyond. If we accept this argument, further philosophical or experimental positioning is required to then determine which percentage of the population we should expect to be non-compliers. Arguably, this number will always be somewhat arbitrary.

It might also be interesting to model other rules that are subtly different from rule utilitarianism. Instead of looking at the impact of a certain type of volition when all or most agents choose to perform it, we could look at the average weight of a particular volition in other types of cases. For instance, while rule utilitarianism finds that lying is wrong on the basis that, if everyone lied, utility

would be minimised, it is possible that if only a certain percentage of the population ever lied in a specific set of situations, utility would be maximised and would lead to a better world than a world where there are no liars. We might also simply want to investigate the impact of lying in a world where there are many liars, or across all types of worlds with varying populations of liars. Modelling rules of this kind would enable us to appraise the role of volitions in varying environments, and begin investigating dynamics of cooperation and competition. We could also study some properties of interactions between volitions, by for instance looking at the average effect of pairs or groups of volitions. For example, giving an offender an alternative sanction under methods of restorative justice might only have long term beneficial effects if it is also accompanied by other initiatives, such as psychological support or victim-offender mediation. The rule utilitarian may then be interested in finding out whether the universalisation of those enterprises are utility maximising as a group rather than in isolation. This will materialise in the fact that rules become more complex and more situation-dependent. The rule "implement alternative sanction" will become "implement alternative sanction in the presence of psychological support."

Because it is concerned with the aggregate Good, rule utilitarianism is often criticised for carrying the risk of being unfair towards certain groups. Strongly advantaging 90 people while strongly disadvantaging 10 might result in maximum utility where equitable distribution of individual welfare would not. The dynamics of a particular environment might also mean that specific groups will suffer more than others: minority groups whose interests clash with the majority will repeatedly be at a disadvantage, as will outcasts or people with less common priorities. The concept of maximum utility, in itself, does not carry a demand for equality or equity. In terms of our model, we could therefore complexify this notion in the *model of the Right*, or we could make adjustments to a theory of the Good by only considering certain members of identified groups, performing a kind of positive discrimination.

We now turn to deontological theories of the Right. Modelling them will typically be less computationally demanding than for consequentialist ones, since for the most part they do not demand reasoning about causes and consequences but focus on intrinsic features.

9.4 Deontological Ethics

In this section, we present three deontological accounts, two of which are purely deontological – those relating to codes of conduct and to Kantian ethics – and one which includes consequentialist constraints – the doctrine of double effect. Because these principles pertain to intrinsic features of volitions rather than to possible outcomes, we model these theories *in hindsight* rather than *in foresight*. This means that we appraise each volition within the simulation it occurs in, and not

through the prism of other possible simulations. With one exception (dde3), there is no talk of uncertainty or risk, as these do not impact on deontological judgements of permissibility. Moreover, as noted previously, generalising the application of these principles from a volition to an entire scenario will be done in the same way for all deontological theories: a scenario as a whole is impermissible if it contains at least one impermissible volition. The integration rules for all three principles are therefore:

```
imp(_,S):-imp(_,S,T,I).
per(_,S):-sim(S),not imp(_,S).
```

9.4.1 *toR:* Codes of Conduct

> *"Now if I carry out this oath, and break it not, may I gain for ever reputation among all men for my life and for my art; but if I transgress it and forswear myself, may the opposite befall me." Hippocratic Oath, [55]*

A code of conduct is a set of rules which outlines the obligations, prohibitions or responsibilities of an individual, group or organisation. It specifies the principles that guide the decision-making or procedures of those constrained by the code. Codes of conduct vary in their scope and nature, ranging from professional deontological codes to religious commandments. Behaviour and morality is typically determined by an overarching body, such as a company, a state, or God. We here exemplify this kind of ethical constraint by modelling a commonly stated rule which is the prohibition of killing. Such a rule is for instance found in the Declaration of Geneva of the World Medical Association in the form of the statement "I will maintain the utmost respect for human life" [14], or in the Decalogue as the commandment "Thou Shalt Not Kill" (Exodus 20:1-21). We model a rule of this kind by stating that an action is impermissible in so far as it causes what is prohibited -here, killing.

```
imp(conduct,S,T,A):-action(A),r(S,causes,A,T,E),effect(E,neg(life)).
per(conduct,S,T,A):-action(A),occurs(S,A,T),not imp(conduct,S,T,A).
```

We can then vary these kinds of rules to fit with all kinds of moral codes, which might also pertain to causal relations other than causing, such as enabling other people's actions. We might cite the legal concepts of aiding and abetting, which capture in English law the crime or felony that consists in helping someone else perform a crime or felony. Integrating omissions also presents the opportunity to model variations of the kind found between "do not kill someone" and "do not let someone die." It simply requires that we replace the actions in the rules by omissions or by volitions as a whole, depending on the requirement.

Codes of conduct can be seen to closely resemble theories of the Good, in that they make judgements that are ultimately based on the way in which volitions impact on independent good and bad entities. Killing is prohibited because it takes a life, it infringes on the right to life. But they differ in that they make different demands on the targeted prohibitions or obligations. For example, going by the modelled code of conduct rule, killing is forbidden even if it saves other lives. But the consideration of the killing as constitutive of the Bad can factor into a wider reasoning structure comprising of various modalities and overarching theories of the Right.

9.4.2 *toR:* Formula of the End in Itself

> *"Act in such a way that you always treat humanity, whether in your own person or in the person of any other, never simply as a means, but always at the same time as an end." I. Kant, 1785 [130]*

The formula of the end in itself is one element of the great breath of Kantian ethics which places special emphasis on the intrinsic value of human life. It is a moral imperative which proscribes using people as means to other ends, for people are ends in themselves in virtue of their very nature as rational beings [130] (depending on our view of person-hood and intrinsic worth, we might also want to include animals as relevant agents to treat as ends in themselves). The formula contrasts intrinsic value, which is persistent and sovereign, with instrumental value, which is dependent upon what it produces. Money might accordingly be seen as having instrumental value, and happiness intrinsic value.

Presumably, aims as discussed by Kant are good aims, so that aiming to harm someone for the sole purpose of harming them is not considered permissible even though the victim is treated not as a means but as an end. We might even consider that harming someone can never count as an instance of treating them *as an end*, by taking a normative view of "ends" as necessarily commanding beneficial impact. In addition, and although this can be debated, we also consider that the formula implies that impacting someone in a positive way but as a means to another end is also impermissible, since it still fails to consider them as persons to the right extent. Helping a poor person in order to look good among one's peers might be considered one such impermissible act. As such, since we concede that causing someone harm (whether as a means or as an end) and impacting on someone as a means to an end (whether in a positive or negative way) are individually sufficient to violate the formula, we define two corresponding rules of impermissibility.

The first rule states that an action is impermissible if it causes an event which brings harm to a given person or group. This is identified by the `impact(E,G,neg(M),N)` in which a – systematically beneficial – modality is negated, with the person or set of people denoted by G. The second rule

states that an action is impermissible if it causes an event which has an impact on a given target or group but where that event is not an aim of the action. Here, the `impact(E,G,M,N)` indicates that at least one person is impacted by E, and the `aim(A,E)` predicate indicates the aim E of A. For example, stealing is forbidden because the aim of stealing is for the thief to become richer, but it also causes the victim to be poorer and upset, even though this is not the aim of the stealing action. Note that action aims must be declared for the rule to function successfully. If the modeller fails to state an aim for an action, it will automatically be deemed impermissible. In addition, because of the reflexivity of causes, these two rules also apply to the action itself (and not just to the other events it causes). An action is permissible if it is not found impermissible.

```
imp(kant,S,T,A):-r(S,causes,A,T,E),impact(E,G,neg(M),N),action(A).
imp(kant,S,T,A):-r(S,causes,A,T,E1),impact(E1,G,M,N),not aim(A,E1),action(A).
per(kant,S,T,A):-action(A),occurs(S,A,T),not imp(kant,S,T,A).
```

The formula of the end in itself is sometimes seen as an application of Kant's first formulation of the categorical imperative, the formula universal law, which states that "I ought never to act except in such a way that I could also will that my maxim should become a universal law" [130]. Consider the following. Someone who rejects the formula of the end in itself abides by a rule of the form "when I wish to, I may use other people solely as means" [54]. This is a rule which cannot be universalised under the requirement of the categorical imperative, for it would imply that the agent wills others to treat him as a means to their own ends. In so far as he considers himself to be a valuable rational being and an end in itself, this is inappropriate. As such, anyone abiding by the formula of the end in itself abides by the formula of universal law. More generally, as discussed by Kant, the formula of universal law can be understood as a test of logical consistency. Any action which fails it is an action whose universalisation *logically undermines* the very concept of the action. Two striking examples of this are lying and stealing. If everyone were to lie, then lying itself would be undermined, since no one would ever believe anyone else and therefore a lie would not be a viable action. Likewise, stealing, say by borrowing money and not giving it back, would be undermined by universalisation: if it became generalised, then no one would lend anyone else anything. But reasoning through the categorical imperative implies a complex representation of intentionality as well as group behaviour, including epistemic dynamics relative to individual and shared knowledge, commonsense grounding and communication. Because our framework is not yet developed far enough to appraise these, we here refrain from attempting to model this second formula.

9.4.3 *toR:* The Doctrine of Double Effect

> *"Nothing hinders one act from having two effects, only one of which is intended, while the other is beside the intention." T. Aquinas, 1485 [8]*

As introduced in section 1.2.1, the doctrine of double effect (DDE) is a set of ethical criteria employed for evaluating the ethical permissibility of an action that has both good and bad consequences [74]. It allows that while actions with negative consequences are morally prohibited a *priori*, there may be instances in which they are morally permissible. The first mention of the doctrine of double effect is credited to Thomas Aquinas amid his discussion of the permissibility of self-defence in the *Summa Theologica* (II-II, Qu. 64, [8]). Against Augustine who saw no justification for it, he claimed that killing in self-defence is morally justifiable as long as the killing is not intentional. He argued that an action can have two consequences, one intended and the other "*beside the intention*." Concerning self-defence, the intended action would consist in saving one's life, and the unintended one would consist in killing the aggressor. Aquinas further provides a general clause of proportionality, suggesting that a lawful act of self-defence may be rendered unlawful if unnecessary violence is used against the aggressor: a disproportional response would be akin to intentionality.

Later versions of the doctrine have generalised the principle to encompass all cases where a distinction can be drawn between causing a morally grave harm as a side effect of pursuing a good end and causing a morally grave harm as a means to a good end. It is often discussed relative to situated political decisions, such as wartime conduct. For example, it will permit the destruction of a military facility to put the enemy out of action even if it affects civilians, but condemn the direct bombing of civilians for purposes of intimidation. It is pertinent to historical cases, we might for instance use it to interpret the bombings of Nagasaki and Hiroshima as instances of reaching a good event (the end of the war) by means of a bad one (the killing of innocents). It is also relevant to law. It can be used to explain certain degrees of wrongful doing, such as the difference between murder and manslaughter. The first occurs when the perpetrator directly intends the death of the victim, the second occurs when the perpetrator foresees the death as a possible consequence of a less harmful, but nevertheless blameworthy intention.

As such, the doctrine of double effect can play both a descriptive role in explaining human intuitions and a normative one in guiding law [139]. Mikhail has suggested that it can assume these two roles because, though people might not be aware of the principles that underlie their moral judgements, they are in effect "intuitive lawyers who implicitly recognise the relevance of ends, means, side effects and prima-facie wrongs, such as battery, to the analysis of legal and moral problems" [171]. This was corroborated by experimental findings where subjects were demanded to judge theoretical moral dilemmas and significantly condemned actions which breached the doctrine over those that

didn't, given the same outcomes in consequentialist terms [171]. Finally, we note that the principle is prominent in the Catholic tradition where a strong emphasis is placed on prohibiting causing a harm as a means of pursuing a good end, and has sometimes been used as a more or less convincing argument against abortion and euthanasia. We now turn to our computational approach to the doctrine, looking at each clause in turn. Because it explicitly appraises actions but says nothing of omissions, we limit the model to the judgement of actions only.

The Nature-of-the-act Condition *The act itself must be morally good or at least indifferent.* The first axiom of the DDE is modelled by appealing to the `bad(A,M,G,N)` predicate elaborated in the *model of the Good*. The permissibility of the action relative to this condition therefore directly stems from the agent's implemented theory of the Good. Moreover, appealing to the `bad` predicate rather than to a negative `weight` value means that an action is forbidden as soon as it has a bad element, even if it is also partly good.

```
imp(dde1,S,T,A):-occurs(S,A,T),bad(A,M,G,N),action(A).
```

The Right-intention and Means-end Conditions *The agent may not positively will the bad effect but may merely permit it. The good effect must be produced directly by the action, not by the bad effect.* These two axioms are collapsed into one rule within our model, for we consider that using an event as a means to an end (i.e. for the occurrence or prevention of another event), is equivalent to intending that event. Correspondingly, unintended side effects cannot be means to ends. It follows that the two axioms are analogous computationally, unless intentions are explicitly modelled, which is not the case here. A good effect may be reached in one of two ways, either by causing a desirable event or by preventing an undesirable one. The reverse is true for bad effects. Therefore, four rules must be specified. It is relatively straightforward to model the two cases in which the bad event used a means to a good end is *caused* by the considered action. However, when a good event used a means is *prevented*, the computation is more complicated. In order to postulate that preventing an event led to a good outcome, we must be able to reason over the hypothetical scenario in which the event does occur, to be able to say that it would have led to a bad outcome.

```
imp(dde2,S,T1,A):-
    r(S,causes,A,T1,E1),r(S,causes,E1,T2,E2),weight(E1,N1),N1<0,weight(E2,N2),
    N2>0,action(A).
imp(dde2,S,T1,A):-
    r(S,causes,A,T1,E1),r(S,prevents,E1,T2,E2),weight(E1,N1),N1<0,weight(E2,N2),
    N2<0,action(A).
```

```
imp(dde2,S1,T1,A):-
    r(S1,prevents,A,T1,E1),r(S2,causes,O,T1,E1),r(S2,causes,E1,T2,E2),
    not canArise(S1,E2),sameHistory(T,S1,S2),weight(E1,N1),N1>0,weight(E2,N2),
    N2<0,T1<T2,action(A),omission(O).
imp(dde2,S1,T1,A):-
    r(S1,prevents,A,T1,E1),r(S2,causes,O,T1,E1),r(S2,prevents,E1,T2,E2),
    canArise(S1,E2),sameHistory(T,S1,S2),weight(E1,N1),N1>0,weight(E2,N2),
    N2>0,T1<T2,action(A),omission(O).
```

In the third and fourth rules, the `r(S2,causes,O,T1,E1)` requirement serves to ensure that no extra action occurs between T1 and T2 in S2, meaning that we only consider the simulation in which the affector action does not occur and nor does any other action until the time of occurrence of E1. We do not want to allow for additional actions to impact on the causal relations. These rules therefore state that A prevents a good event which, in the absence of A and with all other things kept equal, would have occurred and led to a good end-state (either a caused good or a prevented bad) that would not have been true in the presence of A. An example of this type of relation could be found in an agent who prevents a medic from saving a criminal's life. The DDE would preclude this action because it would involve doing harm to the criminal in order to do good to those the criminal would harm otherwise.

The Proportionality Condition *The good effect must be sufficiently desirable to compensate for the allowing of the bad effect.* The last axiom of the DDE introduces a consequentialist requirement, as it demands the weighing against each other of the action's good and bad effects. In order to model it, we simply appeal to the principle of benefits v. costs, so that if the action is impermissible relative to this principle it is also impermissible relative to the DDE. We finally state that a volition I occurring at T in S is impermissible relative to the DDE if it is deemed impermissible by at least one of the doctrine's clauses. We are then left with no, one or a number of permissible actions.

```
imp(dde3,S,T,A):-imp(benefitsCosts,S,T,A),action(A).
per(dde,S,T,A):-
    occurs(S,A,T),not imp(dde1,S,T,A),not imp(dde2,S,T,A),not imp(dde3,S,T,A),action(A).
```

9.4.4 Proof of Concept #6 (*trolley*)

The Trolley Problem

In order to implement the DDE, we appeal to the Trolley problem. As we recall, it goes as follows.

> (*switch*) A train is running towards five people stationed on train tracks; these people are workmen repairing the track. If the agent does nothing, the train will run over and kill them. However, the agent has the option of actioning a switch that will deviate this train from these tracks and place it onto another set of tracks along which one person is walking. This will kill the person.

Intuitively, respondents tend to agree that this action (actioning the switch), is ethically admissible [222]. This fits with the utilitarian notion that killing one person to save five is the better option -the other one being no action at all. Now take another case,

> (*push*) There is no switch button, instead there is a bridge above the train tracks on which stands an onlooker. Here, the agent knows that if they push the onlooker onto the tracks, the train will run into and kill the onlooker, and stop as a result of the crash, thereby saving the five workmen.

Intuitions here diverge from the previous example as people significantly tend to consider the push action ethically impermissible [222]. Here, responses are motivated by something other than utilitarian reasoning, as they choose the death of five people over the death of one. The DDE succeeds in justifying these seemingly inconsistent intuitions. In (*push*), while the nature-of-the-act and proportionality conditions are met, the means-end and right-intention conditions are violated: the death of the onlooker is used as a means to preventing the death of the five workmen, as such it is not just a foreseen side-effect but an intended consequence. Reversely, in (*switch*), the death of the one person walking on the other tracks plays no upstream causal role in the saving of the five: the five are saved whether or not that one person dies, as long as the train leaves its original tracks. As such, by drawing a distinction between intending harm and merely foreseeing it, the doctrine can justify departures from purely consequentialist thinking. We model the dilemma in the following way.

```
% -------------- Scenario Generation
sim(simName).
0{performs(simName,A,T):action(A)}1:-time(T).
% -------------- Domains
number(-100..100).
time(0..6).
agent(decider;bystander;onlooker;workers).
object(G):-agent(G).
nonInertial(on(onlooker,bridge(M));near(G,M)):-track(M),agent(G).
positiveFluent(on(train,M);on(J,M);on(J,bridge(M));near(G,M)):-
```

```
    track(M),object(J),agent(G).
track(t(L,H)):-lanes(L),length(H).
lanes(main).
lanes(side).
close(main,side).
length(0..5).
buttonOn(t(main,0)).
bridgeOver(t(main,0)).
% -------------- Initial Situation
initially(alive(G)):-agent(G).
initially(near(decider,t(main,0))).
initially(on(train,t(main,0))).
initially(on(workers,t(main,3))).
initially(on(bystander,t(side,1))).
initially(on(onlooker,bridge(t(main,0)))).
% -------------- Event Specification
% Act: switch
capable(decider,switch(H,L1,L2)):-buttonOn(t(L1,H)),close(L1,L2).
prec(on(train,t(L1,H)),act(G,switch(H,L1,L2))):-buttonOn(t(L1,H)),close(L1,L2),agent(G).
prec(near(G,t(L1,H)),act(G,switch(H,L1,L2))):-buttonOn(t(L1,H)),close(L1,L2),agent(G).
effect(act(G,switch(H,L1,L2)),neg(on(train,t(L1,H)))):-
    buttonOn(t(L1,H)),close(L1,L2),agent(G).
effect(act(G,switch(H,L1,L2)),on(train,t(L2,H))):-
    buttonOn(t(L1,H)),close(L1,L2),agent(G).
% Act: push
capable(decider,push(J,M)):-bridgeOver(M),object(J).
prec(on(J,bridge(M)),act(G,push(J,M))):-bridgeOver(M),object(J),agent(G).
prec(near(G,M),act(G,push(J,M))):-bridgeOver(M),object(J),agent(G).
effect(act(G,push(J,M)),on(J,M)):-bridgeOver(M),object(J),agent(G).
effect(act(G,push(J,M)),neg(on(J,bridge(M)))):-bridgeOver(M),object(J),agent(G).
% Auto: run
auto(run(train,M)):-track(M).
prec(on(train,M),run(train,M)):-track(M).
effect(run(train,t(L,H)),on(train,t(L,H+1))):-lanes(L),length(H).
effect(run(train,M),neg(on(train,M))):-track(M).
% Auto: crash
auto(crash(J,M)):-object(J),track(M).
prec(on(J,M),crash(J,M)):-object(J),track(M).
```

```
prec(on(train,M),crash(J,M)):-object(J),track(M).
effect(crash(J,M),neg(on(train,M))):-object(J),track(M).
effect(crash(J,M),neg(life)):-object(J),track(M).
effect(crash(J,M),neg(on(J,M))):-object(J),track(M).
% Priorities
priority(I,U):-volition(I),auto(U).
priority(crash(J,M),run(train,M)):-auto(crash(J,M)).
% --------- Target Specification
targetNumber(workers,5).
targetNumber(overlooker,1).
targetNumber(bystander,1).
targetWeight(G,1):-agent(G).
% --------- Modality Specification
% rights
right(life).
modalityWeight(life,10).
% --------- Ethical Event Specification
impact(crash(G,M),G,neg(life),100):-agent(G),track(M).
```

This domain allows the *decider* agent to choose between three volitions at t=0: switching the train onto the side tracks, pushing the onlooker or doing nothing. The onlooker is on the bridge, the bystander is on the first section of the side tracks and the group of workers is on the third section of the main tracks. In order for the *switch* action to exist somewhere along the tracks, regardless of whether it is complete, there must be a button at that location. Similarly, there must be a bridge over the tracks for the *push* action to exist. There are two automatic events: *run*, which enables the train to go forwards, and *crash*, which occurs when the train runs into a physical object (human or other). We define a theory of the Good based on rights such that a crash with a person kills the person and violates the right to life. This right is given the arbitrary value of 10 for all persons, so that no death is better or worse than any other. Because there are 5 workers, a crash with them will weigh 50.

A selective look at the answer set demonstrates that three simulations are generated, one for each possible volition (given the Trolley dilemma, the agent cannot choose more than one volition per simulation). Because we do not apply it to omissions, only actions are appraised through the DDE even though an omission is modelled. We find that the *switch* action is permissible while the *push* action is not. This results from the fact that *push* causes the crash with the onlooker (a bad means) which prevents the crash with the workers (a good end-state), whereas *switch* both causes the crash with the bystander and prevents the one with the workers while there is no causal relation

between these two crashes. We can contrast these results with the ones given by consequentialist principles. For example, the principle of benefits v. costs deems that both *switch* and *push* are permissible while omitting is not, since they each lead to one death while omitting leads to five. This example specifically focuses on the 2nd and 3rd conditions of the DDE. But we can easily play with the features of the planning domain to test the reactivity of the rules to situational changes. For example, if we were to place the workers on the side tracks and the bystander on the main tracks, the switch action would become impermissible in view of the proportionality condition.

```
% s(1)
occurs(s(1),omit(decider,0),0).
occurs(s(1),run(train,t(main,0)),1).
occurs(s(1),run(train,t(main,1)),2).
occurs(s(1),run(train,t(main,2)),3).
occurs(s(1),crash(workers,t(main,3)),4).
% s(2)
occurs(s(2),act(decider,switch(0,main,side)),0).
occurs(s(2),run(train,t(side,0)),1).
occurs(s(2),crash(bystander,t(side,1)),2).
% s(3)
occurs(s(3),act(decider,push(onlooker,t(main,0))),0).
occurs(s(3),crash(onlooker,t(main,0)),1).
% Theory of the Good
bad(crash(bystander,t(side,1)),life,bystander,10).
bad(crash(onlooker,t(main,0)),life,onlooker,10).
bad(crash(workers,t(main,3)),life,workers,50).
% Ethical judgement: switch
per(dde,s(2),0,act(decider,switch(0,main,side))).
per(benefitsCosts,s(2),0,act(decider,switch(0,main,side))).
r(s(2),causes,act(decider,switch(0,main,side)),0,crash(bystander,t(side,1))).
r(s(2),prevents,act(decider,switch(0,main,side)),0,crash(workers,t(main,3))).
% Ethical judgement: push
imp(dde2,s(3),0,act(decider,push(onlooker,t(main,0)))).
per(benefitsCosts,s(3),0,act(decider,push(onlooker,t(main,0)))).
r(s(3),causes,act(decider,push(onlooker,t(main,0))),0,crash(onlooker,t(main,0))).
r(s(3),prevents,crash(onlooker,t(main,0)),1,crash(workers,t(main,3))).
% Ethical judgement: omit
imp(benefitsCosts,s(1),0,omit(decider,0)).
```

Variations on the Trolley Problem

A number of variations to the Trolley problem have been put forward by researchers in both the domains of philosophy and experimental psychology in order to further describe the nature and origins of moral knowledge and intuition. One approach termed Universal Moral Grammar (UMG) suggests that an "adequate scientific theory of moral cognition will often depend more on the computational problems that have to be solved than on the neurophysiological mechanisms in which those solutions are implemented" [163]. It points to an innate human faculty to make moral judgements, shaped later by appropriate experience. The role of the Trolley problem variations is therefore to test whether individuals display stable and systematic intuitions regarding moral issues, based on some particular rules, concepts or principles [171]. This was already the case with the original Trolley problem, but these examples provide a greater number of nuances. As discussed in the introduction, creating new dilemmas opens avenues for testing ethical theories and refining them.

The Loop This case was put forward by Judith Jarvis Thomson in 1985 [236]. It takes the original (*switch*) case in which the trolley is headed toward five workmen but can be redirected onto side tracks where one innocent bystander stands, and adds that the side tracks loop back toward the five. Therefore, if it were not the case that the trolley would hit the one and stop, it would go around and kill the five. If the agent throws the switch, the bystander on the side tracks dies, if not, the five workmen die (note that if the five were not present, the trolley would not go around and hit the one the other way around, but would carry on harmlessly down the track). It is similar to the original switch example but differs in that the death of the bystander – if the switch is thrown – prevents the death of the five. According to the DDE, flipping the switch becomes impermissible, for it violates the means-end condition. The case is modelled by taking the original switch case and adding the rule:

```
effect(run(train,t(side,2)),on(train,t(main,2))).
```

The outcome, as expected, is that the *switch* action is impermissible.

```
(..)
r(s(2),causes,act(decider,switch(0,main,side)),0,crash(bystander,t(side,1))).
r(s(2),prevents,crash(bystander,t(side,1)),2,crash(workers,t(main,3))).
```

This case presents a challenge to the DDE in that it is difficult to justify why the original switch action is permissible where this one is not. The existence of additional tracks seems insufficient to throw ethical judgement. This suggests that our intuitions might be explained by factors beyond the

intend/foresee distinction. In order to address it, Frances Kamm has suggested that a distinction can be drawn between acting *because-of* and acting *in-order-to*. She argues that what she calls the doctrine of triple effect (DTE) can explain the permissibility of redirecting the trolley in this case. The idea is that, beyond the intending/foreseeing distinction of DDE, there is a significant difference between doing something because an effect will occur and doing it in order that it occurs, *whereby doing something because an effect will occur does not imply that one intends that the effect occurs.* She gives the following example to make clear the distinction, as described in [236].

> (*party*) We intend to throw a party in order to have fun. We foresee though that this will result in a big mess, and we will not have a party if we will be left to clean up that mess by ourselves. However, we foresee that if we throw the party, our friends will feel indebted to us and this will cause them to help clean up. Hence, we have the party because we believe that our friends will feel indebted and because we will not have a mess to clean up. However, we do not give the party in order to make our friends feel indebted or to clean up for us.

She therefore argues that in the loop case, the agent redirects the trolley because he believes that the one will be hit, but not in order to hit the one. This means that hitting the one is not *intended* and is therefore permissible. Fundamentally, Kamm suggests that the DTE is justified by the distinction between primary and secondary reasons for action [236]. An agent intends an end-state ψ by performing an affector ϕ if and only if ψ is the agent's primary reason for doing ϕ. In the party example, the primary reason is to have fun and the secondary reason comes from the knowledge that the undesirable effect of throwing the party will be taken care of by the foreseen evil (friends feeling indebted) that the agent produces (by choosing to have the party): "The bad effect we might say, defeats the defeaters of my primary reason, and so maintains the sufficiency of my primary (goal) reason. It is not, however, my goal in action to produce what will defeat the defeaters of my goal" [129].

In order to apply this to (*loop*), we begin by identifying the possible outcomes of the case. The five are subject to two possible threats. First, the trolley can hit them from the main tracks. Second, by flipping the switch, the agent creates another threat, that of the trolley hitting them from the other direction, coming from the side tracks. Therefore, we can identify the primary reason as *preventing the trolley to hit the five from the main tracks*, while the secondary reason emanates from knowing that *the new danger for the five created by redirecting the trolley to the side tracks can be taken care of by the foreseen evil of hitting the one.* Following this reasoning, because our primary reason is to avoid the first hit with the five, and is not directly concerned with producing the effect of hitting the one bystander, we conclude that killing the one is not intended. However, attempting to model

the DTE makes a difficulty emerge: why should avoiding a crash coming from the main tracks be a primary reason while avoiding a crash coming from the side tracks be secondary? Is there any fundamental ethical difference between the two? Can we then translate it into predicate form? It seems not. This brings us to the wider question about whether we can find a formal and general rule for distinguishing primary from secondary reasons, and distinguishing doing something *in-order-to* from doing it *because-of*. As it stands, we could not. This difficulty in modelling therefore led us to question the validity and applicability of the DTE in agreement with [149], though further work on this theory might bring more enlightenment. Though there exists a great number of other Trolley problem variations, we end the proof of concept by modelling another two.

The Man in Front Here, as in (*loop*), the side tracks also loop back into the main tracks where the five workmen are. The agent can switch the train onto the side tracks where there is a heavy rock, which will stop the train in case of collision. In addition, there is a man standing on the side tracks in front of this rock, and a collision between him and the train would result in his death but not in the train stopping. If the agent throws the switch, the bystander on the side tracks dies, if not, the five workmen die. In addition to the added rule of the (*loop*) case, we add the presence of rocks, take out the fact that a crash with the person stops the train, while considering that a crash with the rocks does. Because of the priority of `crash` over `run`, we also need to restart the train after the crash with the person occurs.

```
object(rocks).
initially(on(rocks,t(side,2))).
effect(crash(bystander,t(side,H),on(train,t(side,H)))):-length(H).
effect(crash(J,M),neg(on(train,M))):-object(J),track(M),J!=bystander.
[take out: effect(crash(J,M),neg(on(train,M))):-object(J),track(M).]
```

The result, as expected with the DDE, is that the *switch* action is permissible. The death of the bystander plays no causal role in the prevention of the crash with the workers, since it does not stop the train. It is a side effect of preventing the death of the workers by colliding with the rocks. This case presents the interesting feature that the side effect (death of the bystander) occurs before the intend event (colliding with the rock).

```
(..)
occurs(s(3),act(decider,switch(0,main,side)),0).
occurs(s(3),run(train,t(side,0)),1).
occurs(s(3),crash(bystander,t(side,1)),2).
occurs(s(3),run(train,t(side,1)),3).
```

```
occurs(s(3),crash(rocks,t(side,2)),4).
per(dde,s(3),0,act(decider,switch(0,main,side))).
r(s(3),causes,act(decider,switch(0,main,side)),0,crash(bystander,t(side,1))).
r(s(3),causes,act(decider,switch(0,main,side)),0,crash(rocks,t(side,2))).
r(s(3),prevents,crash(rocks,t(side,2)),4,crash(workers,t(main,3))).
r(s(3),prevents,act(decider,switch(0,main,side)),0,crash(workers,t(main,3))).
```

The Collapsing Bridge In this case, we only consider the main tracks. Here the agent can throw a switch that will make a bridge overlooking the tracks collapse onto them and stop the train. There is a man standing on the bridge. If the agent throws the switch, the person originally on the bridge dies; if not, the five workmen do. Note that when the person falls as a result of the bridge collapsing they don't cause a crash, but just die from the fall. We add the following facts and rules, and find, as expected, that the *break switch* action is permissible.

```
object(rocks).
positiveFluent(broken(bridge(M))):-bridgeOver(M).

% Act: breackSwitch
capable(decider,breackSwitch(bridge(M))):-bridgeOver(M).
prec(near(G,M),act(G,breackSwitch(bridge(M)))):-bridgeOver(M),agent(G).
effect(act(G,breackSwitch(bridge(M))),broken(bridge(M))):-bridgeOver(M),agent(G).
effect(act(G,breackSwitch(bridge(M))),on(rocks,M)):-bridgeOver(M),agent(G).

% Auto: fall
auto(fall(G,M)):-bridgeOver(M),agent(G).
prec(on(G,bridge(M)),fall(G,M)):-bridgeOver(M),agent(G).
prec(broken(bridge(M)),fall(G,M)):-bridgeOver(M),agent(G).
effect(fall(G,M),neg(life)):-bridgeOver(M),agent(G).
effect(fall(G,M),neg(on(G,bridge(M)))):-bridgeOver(M),agent(G).
```

```
(..)
per(dde,s(4),0,act(decider,breackSwitch(bridge(t(main,0))))).
r(s(4),causes,act(decider,breackSwitch(bridge(t(main,0)))),0,crash(rocks,t(main,0))).
r(s(4),prevents,act(decider,breackSwitch(bridge(t(main,0)))),0,crash(workers,t(main,3))).
r(s(4),prevents,crash(rocks,t(main,0)),1,crash(workers,t(main,3))).
```

This case is interesting when compared to the original *push* action. It changes it by turning the bad effect into a side effect but also by making the act impersonal. It is impersonal because,

instead of pushing the bystander, the agent presses on a switch which makes the bridge collapse. This personal/impersonal distinction is often advanced when attempting to find other reasons for distinguishing *switch* and *push* than the foresee/intend distinction.

9.5 Ethics Based on Causal Properties

In this section, we discuss a set of ethically significant distinctions which can be efficiently modelled by drawing on the concepts and predicates defined in the *causal model*.

9.5.1 *toR:* The Doctrine of Doing and Allowing

The doctrine of doing and allowing encapsulates many different theories whose common ground is to attempt to justify why the distinction between doing and allowing harm is morally significant. Across the many versions of the doctrine, the very concepts of doing and allowing vary widely, but all have in common the claim that causing is worse than its weaker counter part. We describe five of them here, as found in [176] and [201].

Causing a harm v. Enabling a harm This account distinguishes causing a harm, such as shooting someone, from enabling a harm, such as giving someone a gun for them to shoot someone with.

Casing a harm v. Omitting to prevent a harm This account distinguishes causing a harm, such as drowning a baby, from omitting to prevent a harm, such as omitting to save a drowning baby.

Causing a harm v. Allowing a harm This account distinguishes causing a harm, such as suffocating someone, from allowing a harm, such as disconnecting a respirator machine from someone in need of it. This account focuses on the removal of defences: allowing amounts to removing a defence to the harm.

Causing a harm v. Redirecting a threat This account distinguishes causing a harm, such as starting a car and running over a pedestrian, from redirecting a threat, such as running over a pedestrian in the attempt to prevent another collision.

Causing a harm v. Accelerating a harm This account distinguishes causing a harm, such as pushing someone off a cliff, from accelerating a harm, such as cutting someone off a climbing rope that will break and kill everyone attached to it if the weight is not alleviated somehow.

These distinctions can all be addressed by appealing to the concepts modelled in the present framework.

Causing v. Enabling

This distinction is captured by our differentiation of the concepts of *causing* and *enabling*. An agent enables a harm if it enables an action that leads to a harmful event[4].

```
dda(S,causeHarm,A,T,E):-r(S,causes,A,T,E),weight(E,N),N<0,action(A).
dda(S,enableHarm,A1,T,E):-
    r(S,enables,A1,T,A2),r(S,causes,A2,T,E),weight(E,N),N<0,action(A1;A2).
```

Causing v. Omitting to Prevent

This corresponds to our concepts of *actions* and *omissions*. An agent causes a harm if a harmful event is the consequence of its action. An agent omits to prevent a harm if a harmful event is the consequence of its omission, provided preventing it was possible. This last clause corresponds in our framework to the requirement that this causal link is not extrinsically necessary, that is, there was an available action or set of actions that could have prevented the harm. We model it such,

```
dda(S,omitPrevHarm,O,T,E):-
    r(S,causes,O,T,E),weight(E,N),N<0,notNecessary(S,causes,exc,O,T,E),omission(O).
```

Note, the `notNecessary` requirement allows that the end-state be avoided by any opposing relation, not just by direct prevention. If we wanted to restrict the rule to direct prevention, we could also model the rule by looking for a relation of prevention in another simulation that has the same history as S up to T, between a viable alternative to the omission and the end-state.

Causing v. Allowing

If we unpack the example given to explain this distinction, we find that removing a defence corresponds to preventing something from preventing a harm. An agent that turns off a respirator machine prevents this machine from preventing the patient's death: an agent allows a harm to occur if it prevents an event that would have prevented a harmful event. This corresponds to our definition of `impedes(prevents)`. As we recall, this is,

[4]It should be noted that the rules modelled in this section all have equivalent counterparts, whereby harm is caused or allowed not through *causing bad events*, but through *preventing good ones*. For the sake of clarity and concision we do not describe them in detail here but they are to be modelled in the same fashion, replacing supporting causal relations with opposing ones, and `bad` valuations with `good` ones. In addition, equivalent rules can be modelled to account for beneficial outcomes, differentiating such things as caused blessings from enabled blessings

```
r(S1,impedes(prevents),E1,T1,E3):-r(S1,prevents,E1,T1,E2),r(S2,prevents,E2,T2,E3),(...).
```

But an agent might also remove a defence by excluding the possibility of acting to maintain this defence. Take Boorse and Sorenson's escape case [33].

> A bear charges out of the woods towards me and my friend. I get up to run away. My friend yells at me "fool, you can't outrun a grizzly." I shout back as I take off, "I only have to outrun you."

This is a case of allowing harm because I remove the defence that is between the bear and my friend: myself. In this case, by escaping, I exclude the possibility to help by friend (acting to prevent a harm done to him) or stop the bear (acting to exclude the bear attacking my friend, assuming a bear is enough of a conscious entity to be considered an acting agent). This corresponds to the following two definitions of `impedes(excludes)`,

```
r(S1,impedes(excludes),E1,T1,E3):-r(S1,excludes,E1,T1,E2),r(S2,prevents,E2,T2,E3),(...).
r(S1,impedes(excludes),E1,T1,E3):-r(S1,excludes,E1,T1,E2),r(S2,excludes,E2,T2,E3),(...).
```

We can also imagine a situation in which a defence is removed by preventing an event that would have excluded a harmful action. An agent removes a defence and allows a harm if it unplugs a drip that usually allows the patient to garner enough energy to feed themselves. This correspond to the third definition of `impedes(excludes)`.

```
r(S1,impedes(excludes),E1,T1,E3):-r(S1,prevents,E1,T1,E2),r(S2,excludes,E2,T2,E3),(...).
```

This causing/allowing distinction therefore can be adequately handled by the four cases emanating from the transitivity matrix of preventing*excluding that results in the concepts of *failing to prevent* and *failing to exclude*.

```
dda(S,allowHarm,A,T,E):-r(S,impedes(prevents),A,T,E),weight(E,N),N<0,action(A).
dda(S,allowHarm,A,T,E):-r(S,impedes(excludes),A,T,E),weight(E,N),N<0,action(A).
```

Fittingly, real world instances of allowed harm can be found in techniques of torture under the name of "double prevention techniques." This is the case of sleep deprivation, which sees the interrogator prevent the detainee from sleeping, which then prevents him from maintaining his usual guard or resilience. These techniques are often presented as "torture lite" rather than torture in full, partly because of the role of nature in producing the desired effect.

Causing v. Redirecting

This distinction is captured by the notion of *prevention*. The fact that an agent redirects a threat means that (a) there is an exiting threat upon a party, (b) the agent applies that threat onto another party, (c) the agent removes that threat from the original party. It therefore corresponds to any situation in which an action both causes a harm and prevents another harm. Presumably the two harms will be similar (as in the example of hitting a pedestrian in an attempt to avoid another), though this is not a necessity. I redirect a threat if I break my knee while running away from a robbery. The theory does nevertheless seem to imply that, in order for the redirecting of the threat to be morally acceptable, the original harm and redirected harm be comparable (or else that the redirected harm be weaker than the original). A policeman who shoots someone who is only armed with a taser does presumably more than redirecting a threat. In order to model this, we can for example require that the weight of the two harmful events be close.

```
comparable(E1,E2):-weight(E1,N1),weight(E2,N2),N1-N2<5,N2-N1<5.
dda(S,redirectHarm,A,T,E1):-
    r(S,causes,A,T,E1),r(S,prevents,A,T,E2),weight(E1,N1),N1<0,weight(E1,N1),N1<0,
    comparable(E1,E2),action(A).
```

Causing v. Accelerating

This distinction can be well captured by appealing to the *scenario-based causal* property of counterfactual validity. Indeed, integral to the notion of accelerating is the assumption that the outcome was going to happen anyway, with or without the agent's action. An agent accelerates a harm if it causes a harm that would have occurred at a later time if the action hadn't been performed; this causal relation is counterfactually invalid.

```
dda(S1,accelerateHarm,A,T1,E):-
    notValid(S1,causes,A,T1,E),occurs(S1,E,T2),ovSim(S1,T1,A,S2),occurs(S2,E,T3),
    weight(E,N),N<0,T2<T3,action(A).
```

Doing and Allowing as Moderators of Ethical Judgement

Looking at the answer set from the (*collision*) case, we find, among others, the following lines corresponding to the doctrine of doing and allowing.

```
dda(s(1),omitPrevHarm,omit(iris,0),0,accident(driver)).
dda(s(2),accelerateHarm,act(iris,kill(driver)),0,death(driver)).
dda(s(2),causeHarm,act(iris,kill(driver)),0,death(driver)).
```

```
dda(s(3),allowHarm,act(iris,safety(driver)),0,decline(driver)).
dda(s(4),redirectHarm,act(iris,aid(driver)),1,partRecovery(driver)).
```

Through her inaction at t=0 in s_1, Iris omits to prevent a new accident from occurring, in s_2 she accelerates the driver's death by killing him while at the same time causing this death, in s_3 she allows the driver's health to decline by bringing him to safety and not performing further actions to help him (this is an instance of meta-transitive impeding to exclude), finally, in s_4 she redirects harm by allowing to partly recover (which is bad because it is painful) instead of letting him decline.

We might then integrate these distinctions within the decision-making process of the agent through rules that, for instance, mitigate the weight of events that were allowed rather than caused. We might for example consider that an agent who enables a harm is only half responsible for it. To do this, we might mitigate the calculation of `weightAdded` values by subtracting part of the weights of enabled events (and use this new definition in the rest of the framework), or change the definition of `weightCons` so that it is tailored to different kinds of allowing. Another way of integrating these distinctions within the global ethical architecture can be done by following Michael S. Moore's position that in cases of causing, deontological considerations have precedence, whereas in cases of allowing, consequentialist ones do. This translates into the idea that if I must kill you to save two others, the deontological requirement not to kill prohibits me from doing it, but if I must simply let you die in order to save two others, then the consequentialist calculation that two lives are better than one licenses me to allow it. We can exclude the irrelevant theories from deliberation by creating an `adequate` predicate to be added to the body of the rules of permissibility of all theories of the Right, so that judgements of permissibility will only emanate from appropriate ones. It may take the following form.

```
deontology(conduct;kant;dde).
consequentialism(benCosts;uti;leastBad;pureBad;ruleUti).
adequate(H,S,T,A):-dda(S1,causeHarm,A,T1,E),consequentialism(H).
adequate(H,S,T,A):-dda(S1,allowHarm,A,T1,E),deontology(H).
imp(H,S,T,A):-(...),adequate(H,S,T,A).
per(H,S,T,A):-(...),adequate(H,S,T,A).
```

To conclude, we believe that the distinctions made by philosophers throughout the doctrine of doing and allowing can be remarkably well handled by the concepts developed in this framework. Moreover, appealing to a coherent set of common basic concepts enables us to model these distinctions in what we argue is a more systematic way than what emerges from appealing to complex and

equivocal notions like "defenses" or "redirection." Quoting Frank Jackson as reported by Moor in [176], "the doctrine of doing and allowing is a mess." A confrontation with explicit computational modelling helps to clear it up. We should nevertheless note, and this is also true for the rest of the framework, that modelling these distinctions is much dependent on the way in which we characterise end-states. But it might not always by clear whether a particular one should be translated into a fluent, an occurring event or the non-occurrence of an event. For instance, a death may be an event (dying) or a prevented event (living). It is up to the modeller to determine which is more adequate for the situation at hand.

9.6 Discussion and Related Works

It is fruitful to compare our model with previously proposed approaches and current works in computational ethics. We will focus here on the example of the Trolley problem and the doctrine of double effect for it has been modelled by others using prospective logic. Pereira and Saptawijaya model the situation in which the agent throws the switch as follows [185]:

turnSide ← *consider(throwingSwitch).*
kill(1) ← *human(X), onSide(X), turnSide.*
end(saveMen, niKill(N)) ← *turnSide, kill(N).*
observedEnd ← *end(X,Y).*

In parallel, the case in which the agent pushes a person on the tracks is as follows:

onTrack(X) ← *consider(shove(X)).*
stopTrain(X) ← *onTrack(X), heavy(X).*
kill(1) ← *human(X), onTrack(X).*
kill(0) ← *inanimateObject(X), onTrack(X).*
end(saveMen,iKill(N)) ← *human(X), stopTrain(X), kill(N).*

In order to ascribe ethical criteria, they employ a priori constraints which rule out impermissible actions according to a particular ethical rule (they here correspond to the means-end condition of the DDE) and a posteriori preferences that eliminate those solutions with worse consequences (the proportionality condition). The means end condition, importantly, is obtained via the two rules:

falsum ← *intentionalKilling.*
intentionalKilling ← *end(saveMen,iKill(Y)).*

The difficulty with this kind of formalization is that it directly embeds the moral requirement into the model of the situation by indicating whether the killing is intentional (*iKill(N)*), or not (*niKill(N)*). The program is "told" whether the outcome of the action fits with the ethical rules in

place, through atomic statements of the form:

end(saveMen, iKill(N)) ← *human(X), stopTrain(X), kill(N).*

This is problematic for a number of reasons. First, it does not do justice to the actual reasoning that underpins moral decision making, by failing to represent the articulation of factors that compose an ethical deliberation. Second, because it atomically specifies the ethical character of the situation's outcome (ethical aspects are domain *dependent*), it requires the creation of a different program for each new case. Therefore, even situations that share common features must be modelled independently, as is the case for trolley variations. This is redundant and can also lead to inconsistencies. Because rules lack expressive power, two identical expressions might refer to diverging stories, for example there is nothing in "*human(X), stopTrain(X), kill(N)*" that indicates whether the killing is intentional or not, and as such could be employed in either case (here it is used for the (*push*) case). Moreover, there is no account of causality, such that the action and its consequences are not dynamically linked; the relationship between them is stated rather than inferred. Therefore, no account of ethical responsibility can be discussed on its basis. Finally, models of this kind cannot logically confront ethical theories so as to make explicit their assumptions and give insight into them, nor can it enable us to explore and generate new ethical dilemmas for further testing. Instead, by first establishing an unchanging and ethics-free account of the world atop which we fit changeable ethical restrictions, we allow for generalisation, flexibility and automation. Separating the ethical constraints from the facts of the world is imperative if we are to model general ethical rules instead of performing case by case discrimination that resembles ethical snap judgement more than it does ethical theory.

Chapter 10

Contribution: Interfacing with the User

In order to enable and facilitate interaction between the framework presented here and potential human users, we have devised a programme that can translate the answer sets into readable English. This programme is run in Python and added to the launch script in the form of a new *translation* definition. It explicitates the ASP model's ''reasons" for choosing one action over an other, extracting the output from the ASP environment and parsing it for translation. It necessitates a new kind of predicate, defined in the original ASP programme, that primes the answer set for parsing, dividing information into relevant portions. The virtue of this tool is that it gives non-specialists the possibility to fully understand and interact with the ASP model. It also provides an explanation that may be reused for further modelling and analysis.

The programme works in the following way. In all cases, it finds the volition under scrutiny, the simulation it occurs in and the time it occurs at. When the volition is permissible, it states just that, as well other possible relevant information, such as its compared weight or the utilitarian rule it belongs to. When the volition is impermissible, it states which ethical rules it has violated and also describes some information which serves to justify the judgement, such as the undesirable events for which the volition is responsible, or its added weight. The following are two examples from the python script that encode two possible outcomes for a volition. In the first case, the considered volition is impermissible because it violates the Means End condition of the DDE, in the second, the considered volition is permissible according to rule utilitarianism. The rest of the script can be found online at: https://github.com/FBerreby/Thesis/blob/master/generate.py

```
def translation(input, output):
    RE_ANSWER = re.compile('Answer:\s*(?P<answer>[0-9]+)')
    stop=False
    with open(input, 'r') as file:
with open(output, 'w')as output:
answer = 0
for line in file.readlines():
# [...]
# Half-way predicate for Rule Utilitarian / permissibility:
# p(ruleUti(s1,S,i1,I,t1,T,n1,N,r1,R)):-per(ruleUti,S,T,I),instance(I,R),ruleWeight(R,N).

elif 'p(ruleUti' in line:
newline = line.replace('p(ruleUti','').replace(',',' ').replace('(',' ').replace(')',' ').
    replace('_',' ').replace('.',' ')
sim = re.search('s1(.*)i1', newline)
volition = re.search('i1(.*)t1', newline)
time = re.search('t1(.*)n1', newline)
rw1 = re.search('n1(.*)r1', newline)
rule1 = re.search('r1(.*)', newline)
output.write ("\n \n The volition:"+volition.group(1))
output.write ("\n which occurs in simulation"+sim.group(1))
output.write ("at time"+time.group(1))
output.write ("is permissible according to Rule Utilitarianism because it is an instance
    of the rule"+rule1.group(1))
output.write ("whose rule weight is"+rw1.group(1))
output.write ("and there exists no alternative volition belonging to a rule whose weight
    is greater.")
# [...]
# Half-way predicate for Means-End condition of the DDE / impermissibility:
# i(dde2_cbpb(s1,S,i1,I,e1,E1,e2,E2,t1,T1)):-r(S,causes,I,T1,E1),r(S,prevents,E1,T2,E2),
    weight(E1,N1),N1<0,weight(E2,N2),N2<0,action(I).

elif 'i(dde2_cbpb' in line:
newline = line.replace('i(dde2_cbpb','').replace(',',' ').replace('(',' ').replace(')',' ').
    replace('_',' ').replace('.',' ')
sim = re.search('s1(.*)i1', newline)
action = re.search('i1(.*)e1', newline)
event1 = re.search('e1(.*)e2', newline)
```

```
event2 = re.search('e2(.*)t1', newline)
time = re.search('t1(.*)', newline)
output.write ("\n \n The action:"+action.group(1))
output.write ("\n which occurs in simulation"+sim.group(1))
output.write ("at time"+time.group(1))
output.write ("is impermissible because it violates the Means End condition of the DDE.")
output.write ("\n It causes an undesirable event which itself prevents another
    undesirable event.")
output.write ("\n The caused undesirable event is:"+event1.group(1))
output.write ("\n The prevented undesirable event is:"+event2.group(1))# [...]
if __name__ == '__main__':
    # [...]
    # Translates the result
    translation('final_output.txt', 'english_output.txt')
```

Running the procedure on the (*venture*) case will then for example produce a list of output of the kind:

```
The volition: act g 1  invest
which occurs in simulation s 243 at time 0 is permissible according to Rule
Utilitarianism because it is an instance of the rule invest whose rule weight is 80
```

A launch of the program on the (*trolley*) case will produce output of the kind:

```
The action: act decider push overlooker t main 0
which occurs in simulation s 3  at time 0   is impermissible because it violates the
Means End condition of the DDE.
It causes an undesirable event which itself prevents another undesirable event.
The caused undesirable event is: crash overlooker t main 0
The prevented undesirable event is: crash workers t main 3
```

Part VII

Discussion

Chapter 11

Conclusion

11.1 Summary of Results and Contributions

In this thesis, we have presented a set of modular models based on expressive rules which together enable the representation of a variety of ethical dilemmas and theories. We aimed to *descriptively* reconstruct *normative* theories of ethics, as well as causal properties which pervade people's intuitions and justifications about what is morally right and wrong. Though ours is just one of many possible translations of these philosophical concepts into logical clauses, it has allowed us to reach a greater understanding of the concepts at play, both formally in relation to the predicates and formalisms employed, and in relation to notions that are employed by philosophers and law-makers. This work is interdisciplinary and informs multiple domains: philosophical ethics, computational ethics, logic programming, reasoning about actions and change and the logical and computational representation of causality and responsibility. To the best of our knowledge, it is the first attempt to assimilate actions, omissions and automatic events within plans of actions for modelling ethical decision-making, as well as to inclusively model the multiplicity of causal relations and properties that derive from these. Pertaining to resulting wisdoms and contributions, confronting ethical theories to logical and computational languages has shed light on the importance and difficulty:

- *of accounting for all the ways in which agents might impact the world*, through actions and omissions alike, but also through direct as well as indirect causal relations relating to produced *and* avoided outcomes. This requirement has led us to define three types of events and nine types of direct and transitive *event-based* causal relations.
- *of handling causal paths in order to justify ethical assessments and decision-making.* In particular, working on this framework has exposed the necessity of adequately handling situational

circumstances if a meaningful account of agent responsibility is to be given. This underlines the fact responsibility concerns not just the effect of agent choices, but is to be apprehended from the state of the world itself. While these remarks belong in the realm of common sense for human agents, they are remarkably heavy in repercussion for the modelling of autonomous agents faced with ethical challenges. This requirement has led to define four properties of *scenario-based* causal properties based on the exploration of alternative versions to an original scenario, including counterfactual validity, cruciality, intrinsic necessity and elicited necessity.

- *of investigating the different ways in which ethical principles might be understood* to pertain to individual volitions or to plans of actions, in a way that might or might not involve uncertainty about the future and about the choices of other agents. This requirement has led to the formulation of two methods for ethically appraising scenarios, either *in hindsight* or *in foresight*.
- *of accounting for the varying dynamics that constrain ethical reasoning*, with reference to multiple modalities, scales and ways to evaluate and weigh events. This requirement has led to the investigation of both consequentialist and deontological frames of reference, culminating in the modelling of three theories of the Good and nine theories of the Right, exemplified by multiple illustrative proofs of concept.
- *of allowing for modularity as well as distinguishing the ethics-free account of the world from the ethical rules that might control agents' behaviour within it*, in order to permit flexibility and adaptability. This requirement has motivated our choice of partitioning the framework into four distinct and commutable models, also meaning that it can be coupled with and benefit from different formalisms, as long as their output or input can be projected into predicate form.
- *of making the framework accessible and transparent.* This requirement has led us to propose an agent architecture that is clear and explicit, and readily translatable into other formalisms such as different languages for reasoning about actions and change or other types of logic programming. It also prompted us to provide a tool to translate its output into natural language for interaction with human users.

11.2 Avenues of Future Work

In order to develop this framework, we envision a number of future avenues and discuss the most immediate ones here.

Intentional states First, we believe that the greatest limitation of the present model is its absence of an explicit representation of agent intentions. Yet the handling of intentionality will enable

a more in depth investigation of many ethical theories, including the doctrine of double effect or virtue-based ethics. Modelling intentions will also enable us to develop certain theories of the Good, for instance by giving a more substantial account of ends by distinguishing such things as killing by mistake and killing by murder.

Epistemic states Related to the question of intention is the representation of agent knowledge. We anticipate that integrating epistemic aspects into the architecture would greatly expand its expressiveness, allowing the exploration of ethically weighted agent behaviour such as lying or information sharing.

Collective decision-making A third and natural avenue for future work is the integration of this individual decision framework into collective multi-agent systems, so as to explore its potential to enable cooperation or collective intelligence. This may generate many new issues, such as the characterisation of societies in which many agents with ethical guiding theories cohabit. It may also enable the development of richer models for representing ethical theories which call upon group behaviour to make moral claims, such as rule utilitarianism or Kant's Categorical Imperative, as well as allow for modelling "at-a-time" cases where multiple agents participate together in the production of a given outcome or concepts of collective concern such as equity.

Legal theory Another future avenue concerns the field of law. This domain engenders a significant amount of dilemmas and concepts that have direct ethical impact on the world and whose modelling is susceptible to help refine. One example of such a concept is transferred malice. Consider, for example, a man who plants a bomb in a building, giving a warning to enable the building to be evacuated. He knows it to be virtually certain that explosives experts will enter the building to try to defuse the bomb. They do so, and one of them is killed when the bomb explodes. From these facts, how much difficulty would there be in attempting to prove that he intended to kill the explosives expert or to cause her serious harm? A computational model may assist in determining this. Other potential concepts to be investigated include oblique intent, duress, or inchoate offences.

Appendices

```
def launch_clingo(input_files, output_file):
    input_args = ' '.join(input_files)
    command = "clingo 0 %s >%s" % (input_args, output_file)
    os.system(command)

def conversion(input, output):
    RE_ANSWER = re.compile('Answer:\s*(?P<answer>[0-9]+)')
    stop=False
    with open(input, 'r') as file:
        with open(output, 'w' )as output:
            answer = 0
            for line in file.readlines():
                new_answer = re.search(RE_ANSWER, line)
                if new_answer:
                    answer = new_answer.group('answer')
                elif 'SATISFIABLE' in line:
                    stop=True
                elif not stop:
                    line = re.sub('simName', 's(%s)' % answer, line)
                    line = re.sub('\s\s*', '.\n', line)
                    output.write("%s" % line)

def translation(input, output):
(...)

if __name__ == '__main__':
    # Lanches Clingo for the 1st time
    launch_clingo(input_files=['action.txt', 'collision.txt'],
    output_file='first_output.txt')
    # Transforms the result
    conversion('first_output.txt', 'first_converted.txt')
    # Lanches Clingo for the 2nd time
    launch_clingo(input_files=['first_converted.txt', 'causal.txt', 'good.txt', 'right.txt',
    'show.txt'], output_file='second_output.txt')
    # Transforms the result
    conversion('second_output.txt', 'final_output.txt')
# Translates the result
    translation('final_output.txt', 'english_output.txt')
```

Bibliography

[1] Robert Merrihew Adams. "A modified divine command theory of ethical wrongness". In: (1997).

[2] Larry Alexander and Michael Moore. "Deontological Ethics". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2016. 2016.

[3] Michael Anderson and Susan Leigh Anderson. "GenEth: A General Ethical Dilemma Analyzer." In: *AAAI*. 2014, pp. 253–261.

[4] Michael Anderson and Susan Leigh Anderson. "Machine ethics: Creating an ethical intelligent agent". In: *AI Magazine* 28.4 (2007), p. 15.

[5] Michael Anderson and Susan Leigh Anderson. "Toward ensuring ethical behavior from autonomous systems: a case-supported principle-based paradigm". In: *Industrial Robot: An International Journal* 42.4 (2015), pp. 324–331.

[6] Michael Anderson, Susan Leigh Anderson, and Chris Armen. "MedEthEx: a prototype medical ethics advisor". In: *Proceedings Of The National Conference On Artificial Intelligence*. Vol. 21. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. 2006, p. 1759.

[7] Michael Anderson, Susan Leigh Anderson, and Chris Armen. "Towards machine ethics". In: *AAAI-04 workshop on agent organizations: theory and practice, San Jose, CA*. 2004.

[8] Thomas Aquinas. *Summa theologica*. Xist Publishing, 2015.

[9] John Arbuthnot. *Of the Laws of Chance; Or, A Method of Calculation of the Hazards of Game...* B. Motte and sold by J. Morphew, 1714.

[10] 384-322 BC Aristotle and James Alexander Kerr Thomson. *The Nicomachean Ethics*. Penguin, 1953.

[11] Ronald C Arkin, Patrick Ulam, and Brittany Duncan. *An ethical governor for constraining lethal action in an autonomous system*. Tech. rep. GEORGIA INST OF TECH ATLANTA MOBILE ROBOT LAB, 2009.

[12] Thomas Arnold and Matthias Scheutz. "The "big red button" is too late: an alternative model for the ethical evaluation of AI systems". In: *Ethics and Information Technology* 20.1 (2018), pp. 59–69.

[13] Isaac Asimov. "Runaround". In: *Astounding Science Fiction* 29.1 (1942), pp. 94–103.

[14] World Medical Association et al. "WMA declaration of Geneva". In: *International Journal of Person Centered Medicine* 4.3 (2015).

[15] Alfred J Ayer. "Critique of ethics and theology". In: *Essays on moral realism* (1988), pp. 27–40.

[16] Alfred Jules Ayer. *Language, truth and logic*. Courier Corporation, 2012.

[17] Jonathan Baron. "Nonconsequentialist decisions". In: *Behavioral and Brain Sciences* 17.1 (1994), pp. 1–10.

[18] Jonathan Baron and Ilana Ritov. "Omission bias, individual differences, and normality". In: *Organizational Behavior and Human Decision Processes* 94.2 (2004), pp. 74–85.

[19] Jonathan Baron and Ilana Ritov. "Protected values and omission bias as deontological judgments". In: *Psychology of learning and motivation* 50 (2009), pp. 133–167.

[20] Cristina Battaglino, Rossana Damiano, and Leonardo Lesmo. "Emotional range in value-sensitive deliberation". In: *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2013, pp. 769–776.

[21] T Beauchamp and J Childress. *Principles of Biomedical Ethics*. Principles of Biomedical Ethics. Oxford University Press, 2001. ISBN: 9780195143317.

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

[23] Nuel D Belnap, Michael Perloff, and Ming Xu. *Facing the future: agents and choices in our indeterminist world*. Oxford University Press on Demand, 2001.

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

[25] Jeremy Bentham. *A fragment on government*. The Lawbook Exchange, Ltd., 2001.

[26] Jeremy Bentham. *The collected works of Jeremy Bentham: An introduction to the principles of morals and legislation*. Clarendon Press, 1996.

[27] Jacques Bernoulli and Jacob Bernoulli. *The art of conjecturing, together with Letter to a friend on sets in court tennis*. JHU Press, 2006.

[28] Fiona Berreby, Gauvain Bourgne, and Jean-Gabriel Ganascia. "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. 2017, pp. 96–104.

[29] Fiona Berreby, Gauvain Bourgne, and Jean-Gabriel Ganascia. "Event-Based and Scenario-Based Causality for Computational Ethics". In: *Proceedings of the 17th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2018.

[30] 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*. Springer. 2015, pp. 532–548.

[31] Joseph A Blass and Kenneth D Forbus. "Moral Decision-Making by Analogy: Generalizations versus Exemplars." In: *AAAI*. 2015, pp. 501–507.

[32] Piero Bonatti et al. "Answer set programming". In: *A 25-year perspective on logic programming*. Springer-Verlag. 2010, pp. 159–182.

[33] Christopher Boorse and Roy A Sorensen. "Ducking harm". In: *The Journal of philosophy* 85.3 (1988), pp. 115–134.

[34] Richard B Brandt. "Ethical theory". In: (1959).

[35] Michael Bratman. "Intention, plans, and practical reason". In: (1987).

[36] By Brewers. "Brewer's Dictionary of Phrase & Fable". In: (1970).

[37] Selmer Bringsjord, Konstantine Arkoudas, and Paul Bello. "Toward a general logicist methodology for engineering ethically correct robots". In: *IEEE Intelligent Systems* 21.4 (2006), pp. 38–44.

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

[39] David Owen Brink. *Moral realism and the foundations of ethics*. Cambridge University Press, 1989.

[40] Joanna J Bryson. "Patiency is not a virtue: the design of intelligent systems and systems of ethics". In: *Ethics and Information Technology* 20.1 (2018), pp. 15–26.

[41] Danilo Bzdok et al. "Parsing the neural correlates of moral cognition: ALE meta-analysis on morality, theory of mind, and empathy". In: *Brain Structure and Function* 217.4 (2012), pp. 783–796.

[42] Rudolf Carnap. *Philosophy and logical syntax*. K. Paul, Trench, Trubner, 1935.

[43] Brian F Chellas. *Modal logic: an introduction*. Cambridge university press, 1980.

[44] James F Childress and Tom L Beauchamp. *Principles of biomedical ethics*. Oxford University Press, USA, 2001.

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

[46] Noam Chomsky. "Terror and just response". In: *Terrorism and international justice* (2002), p. 69.

[47] Elisa Ciaramelli et al. "Selective deficit in personal moral judgment following damage to ventromedial prefrontal cortex". In: *Social cognitive and affective neuroscience* 2.2 (2007), pp. 84–92.

[48] Nicolas Cointe. "Jugement éthique pour la décision et la coopération dans les systèmes multi-agents". PhD thesis. Lyon, 2017.

[49] Nicolas Cointe, Grégory Bonnet, and Olivier Boissier. "Ethical Judgment of Agents' Behaviors in Multi-Agent Systems". In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2016, pp. 1106–1114.

[50] Nicolas Cointe, Grégory Bonnet, and Olivier Boissier. "Multi-agent based ethical asset management". In: *1st Workshop on Ethics in the Design of Intelligent Agents*. 2016, pp. 52–57.

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

[52] André Comte-Sponville. *Le capitalisme est-il moral?* Albin Michel, 2012.

[53] Vincent Conitzer et al. "Moral Decision Making Frameworks for Artificial Intelligence." In: *AAAI*. 2017, pp. 4831–4835.

[54] Neil Cooper. "The Formula of the End in Itself". In: *Philosophy* 63.245 (1988), pp. 401–402.

[55] Hippocrates of Cos. *The Oath*. Loeb Classical Library, 1923.

[56] Natalia Criado et al. "Towards a normative BDI architecture for norm compliance". In: *COIN@ MALLOW2010* (2010), pp. 65–81.

[57] Gordana Dodig Crnkovic and Baran Çürüklü. "Robots: ethical by design". In: *Ethics and Information Technology* 14.1 (2012), pp. 61–71.

[58] Chiara Cumbo, Salvatore Iiritano, and Pasquale Rullo. "Olex–a reasoning-based text classifier". In: *European Workshop on Logics in Artificial Intelligence*. Springer. 2004, pp. 722–725.

[59] Fiery Cushman. "Action, outcome, and value: A dual-system framework for morality". In: *Personality and social psychology review* 17.3 (2013), pp. 273–292.

[60] Fiery Cushman, Liane Young, and Marc Hauser. "The role of conscious reasoning and intuition in moral judgment: Testing three principles of harm". In: *Psychological science* 17.12 (2006), pp. 1082–1089.

[61] Fiery Cushman et al. "Simulating murder: The aversion to harmful action." In: *Emotion* 12.1 (2012), p. 2.

[62] Morteza Dehghani et al. "MoralDM: A Computational Modal of Moral Decision-Making". In: *Proceedings of the 30th Annual Conference of the Cognitive Science Society (CogSci)*. Citeseer. 2008.

[63] Marc Denecker, Lode Missiaen, and Maurice Bruynooghe. "Temporal reasoning with abductive event calculus". In: *Proceedings of the 10th European Conference on Artificial Intelligence, ECAI92*. John Wiley and Sons. 1992, pp. 384–388.

[64] Peter DeScioli, Rebecca Bruening, and Robert Kurzban. "The omission effect in moral cognition: Toward a functional explanation". In: *Evolution and Human Behavior* 32.3 (2011), pp. 204–215.

[65] Virginia Dignum. "Responsible autonomy". In: *arXiv preprint arXiv:1706.02513* (2017).

[66] Patrick Doherty and Jonas Kvarnström. "Temporal action logics". In: *Foundations of Artificial Intelligence* 3 (2008), pp. 709–757.

[67] Thomas Eiter et al. "Resolving conflicts in action descriptions". In: *Frontiers in Artificial Intelligence and Applications* 141 (2006), p. 367.

[68] Kai Epstude and Neal J Roese. "The functional theory of counterfactual thinking". In: *Personality and Social Psychology Review* 12.2 (2008), pp. 168–192.

[69] Brian Falkenhainer, Kenneth D Forbus, and Dedre Gentner. "The structure-mapping engine: Algorithm and examples". In: *Artificial intelligence* 41.1 (1989), pp. 1–63.

[70] Oriel FeldmanHall et al. "What we say and what we do: the relationship between real and hypothetical moral choices". In: *Cognition* 123.3 (2012), pp. 434–441.

[71] Richard E Fikes and Nils J Nilsson. "STRIPS: A new approach to the application of theorem proving to problem solving". In: *Artificial intelligence* 2.3-4 (1971), pp. 189–208.

[72] Roderick Firth. "Ethical absolutism and the ideal observer". In: *Philosophy and Phenomenological Research* 12.3 (1952), pp. 317–345.

[73] Philippa Foot. "Morality, action, and outcome". In: *Morality and objectivity* (1985), pp. 23–38.

[74] Philippa Foot. "The problem of abortion and the doctrine of double effect". In: (1967).

[75] Jean-Gabriel Ganascia. "Ethical system formalization using non-monotonic logics". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 29. 29. 2007.

[76] Jean-Gabriel Ganascia. "Modelling ethical rules of lying with Answer Set Programming". In: *Ethics and information technology* 9.1 (2007), pp. 39–47.

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

[78] Richard T Garner and Bernard Rosen. "A Systematic Introduction to Normative Ethics and Meta-Ethics". In: (1970).

[79] Michael Gelfond. "Answer sets". In: *Foundations of Artificial Intelligence* 3 (2008), pp. 285–316.

[80] Michael Gelfond and Vladimir Lifschitz. "Action languages". In: (1998).

[81] Michael Gelfond and Vladimir Lifschitz. "Classical negation in logic programs and disjunctive databases". In: *New generation computing* 9.3-4 (1991), pp. 365–385.

[82] Michael Gelfond and Vladimir Lifschitz. "Representing action and change by logic programs". In: *The Journal of Logic Programming* 17.2-4 (1993), pp. 301–321.

[83] Michael Gelfond and Vladimir Lifschitz. "The stable model semantics for logic programming." In: *ICLP/SLP*. Vol. 88. 1988, pp. 1070–1080.

[84] Bernard Gert. "Morality: A new justification of the moral rules". In: (1988).

[85] Gerd Gigerenzer. "Moral satisficing: Rethinking moral behavior as bounded rationality". In: *Topics in cognitive science* 2.3 (2010), pp. 528–554.

[86] James Gips. "Toward the ethical robot". In: (1994).

[87] Clark Glymour and Frank Wimberly. "Actual causes and thought experiments". In: *Causation and explanation* 4 (2007), p. 43.

[88] Lou Goble. "Utilitarian deontic logic". In: *Philosophical Studies* 82.3 (1996), pp. 317–357.

[89] Jesse Graham, Jonathan Haidt, and Brian A Nosek. "Liberals and conservatives rely on different sets of moral foundations." In: *Journal of personality and social psychology* 96.5 (2009), p. 1029.

[90] Joshua D Greene. "Solving the trolley problem". In: *A companion to experimental philosophy* (2016), pp. 173–189.

[91] Joshua D Greene. "The secret joke of Kant's soul". In: *Moral psychology* 3 (2008), pp. 35–79.

[92] Joshua D Greene et al. "An fMRI investigation of emotional engagement in moral judgment". In: *Science* 293.5537 (2001), pp. 2105–2108.

[93] Joshua D Greene et al. "Pushing moral buttons: The interaction between personal force and intention in moral judgment". In: *Cognition* 111.3 (2009), pp. 364–371.

[94] Joshua Greene and Jonathan Haidt. "How (and where) does moral judgment work?" In: *Trends in cognitive sciences* 6.12 (2002), pp. 517–523.

[95] Shane Griffith et al. "Policy shaping: Integrating human feedback with reinforcement learning". In: *Advances in neural information processing systems*. 2013, pp. 2625–2633.

[96] Jonathan Haidt. "The emotional dog and its rational tail: a social intuitionist approach to moral judgment." In: *Psychological review* 108.4 (2001), p. 814.

[97] Jonathan Haidt and Craig Joseph. "Intuitive ethics: How innately prepared intuitions generate culturally variable virtues". In: *Daedalus* 133.4 (2004), pp. 55–66.

[98] Jonathan Haidt, Silvia Helena Koller, and Maria G Dias. "Affect, culture, and morality, or is it wrong to eat your dog?" In: *Journal of personality and social psychology* 65.4 (1993), p. 613.

[99] Ned Hall. "Structural equations and causation". In: *Philosophical Studies* 132.1 (2007), pp. 109–136.

[100] Joseph Y Halpern. "A Modification of the Halpern-Pearl Definition of Causality." In: *IJCAI*. 2015, pp. 3022–3033.

[101] Joseph Y Halpern and Judea Pearl. "Causes and explanations: A structural-model approach. Part I: Causes". In: *The British journal for the philosophy of science* 56.4 (2005), pp. 843–887.

[102] Richard Mervyn Hare. *The language of morals*. 77. Oxford Paperbacks, 1991.

[103] Gilbert Harman and Judith Jarvis Thomson. "Moral relativism and moral objectivity". In: (1996).

[104] Roy Forbes Harrod. "Utilitarianism revised". In: *Mind* 45.178 (1936), pp. 137–156.

[105] John C Harsanyi. "Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility". In: *Journal of political economy* 63.4 (1955), pp. 309–321.

[106] John C Harsanyi. "Rule utilitarianism and decision theory". In: *Erkenntnis* 11.1 (1977), pp. 25–53.

[107] John C Harsanyi. "Rule utilitarianism, rights, obligations and the theory of rational behavior". In: *Papers in Game Theory*. Springer, 1982, pp. 235–253.

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

[109] Marc Hauser. *Moral minds: How nature designed our universal sense of right and wrong*. Ecco/HarperCollins Publishers, 2006.

[110] Marc Hauser et al. "A dissociation between moral judgments and justifications". In: *Mind & language* 22.1 (2007), pp. 1–21.

[111] J Henrich, SJ Heine, and A Norenzayan. "The weirdest people in the world The Behavioral and Brain Sciences 33 (2–3), 61–83; discussion 83–135 (2010) 2. 344 K". In: *Lakkaraju et al* ().

[112] Risto Hilpinen. *Deontic logic: Introductory and systematic readings*. Vol. 33. Springer Science & Business Media, 2012.

[113] Christopher Hitchcock. "The intransitivity of causation revealed in equations and graphs". In: *The Journal of Philosophy* 98.6 (2001), pp. 273–299.

[114] Steffen Hölldobler and Josef Schneeberger. "A new deductive approach to planning". In: *New Generation Computing* 8.3 (1990), pp. 225–244.

[115] Brad Hooker. "Rule-consequentialism". In: *Mind* 99.393 (1990), pp. 67–77.

[116] M Hopkins and J Pearl. "Causality and counterfactuals in the situation calculus". In: *Journal of Logic and Computation* 17.5 (2007), pp. 939–953.

[117] J Horty. "Nonmonotonic foundations for deontic logic". In: *Defeasible deontic logic*. Springer, 1997.

[118] John F Horty. *Agency and deontic logic*. Oxford University Press, 2001.

[119] John F Horty and Nuel Belnap. "The deliberative stit: A study of action, omission, ability, and obligation". In: *Journal of philosophical logic* 24.6 (1995), pp. 583–644.

[120] John Hospers. "Rule-Utilitarianism". In: *Ethical theory: classical and contemporary readings* (1999), pp. 201–210.

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

[122] Leonid Hurwicz. *Optimality criteria for decision making under ignorance*. Tech. rep. Cowles Commission Discussion Paper, Statistics, 1951.

[123] Salvatore Maria Ielpa et al. "An ASP-based system for e-tourism". In: *International Conference on Logic Programming and Nonmonotonic Reasoning*. Springer. 2009, pp. 368–381.

[124] Tomi Janhunen et al. "Unfolding partiality and disjunctions in stable model semantics". In: *ACM Transactions on Computational Logic (TOCL)* 7.1 (2006), pp. 1–37.

[125] Nicholas R Jennings, Katia Sycara, and Michael Wooldridge. "A roadmap of agent research and development". In: *Autonomous agents and multi-agent systems* 1.1 (1998), pp. 7–38.

[126] Richard Joyce. *The myth of morality*. Cambridge University Press, 2001.

[127] Shelly Kagan. "The additive fallacy". In: *Ethics* 99.1 (1988), pp. 5–31.

[128] Shelly Kagan. "The limits of morality". In: (1989).

[129] Frances Myrna Kamm. *Intricate ethics: Rights, responsibilities, and permissible harm*. OUP USA, 2007.

[130] Immanuel Kant. "Groundwork of the Metaphysic of Morals, trans. HJ Paton". In: *New York: Harper & Row* 4 (1964), pp. 420–426.

[131] Immanuel Kant. "On a supposed right to lie from altruistic motives". In: *Critical of practical reason and other writings* (1949), pp. 346–350.

[132] Immanuel Kant. *The metaphysical elements of ethics*. Litres, 2017.

[133] Robert E Kass and Larry Wasserman. "The selection of prior distributions by formal rules". In: *Journal of the American Statistical Association* 91.435 (1996), pp. 1343–1370.

[134] John Maynard Keynes. *A treatise on probability*. Courier Corporation, 2013.

[135] Tae-Won Kim, Joohyung Lee, and Ravi Palla. "Circumscriptive Event Calculus as Answer Set Programming." In: *IJCAI*. Vol. 9. 2009, pp. 823–829.

[136] George P Klubertanz. "The New Catholic Encyclopedia". In: *The Modern Schoolman* 46.4 (1969), pp. 377–378.

[137] Clyde Kluckhohn. *Values and value-orientations in the theory of action: An exploration in definition and classification*. 1951.

[138] Boris Kment. *Modality and explanatory reasoning*. OUP Oxford, 2014.

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

[140] Robert A Kowalski and Fariba Sadri. "The Situation Calculus and Event Calculus Compared." In: *ILPS*. Vol. 94. 1994, pp. 539–553.

[141] Robert Kowalski and Fariba Sadri. "Reconciling the event calculus with the situation calculus". In: *The Journal of Logic Programming* 31.1-3 (1997), pp. 39–58.

[142] Robert Kowalski and Marek Sergot. "A logic-based calculus of events". In: *Foundations of knowledge base management*. Springer, 1989, pp. 23–55.

[143] Bruno Latour. "10 "Where Are the Missing Masses? The Sociology of a Few Mundane Artifacts"". In: (1992).

[144] Joohyung Lee and Ravi Palla. "Reformulating the situation calculus and the event calculus in the general theory of stable models and in answer set programming". In: *Journal of Artificial Intelligence Research* 43 (2012), pp. 571–620.

[145] Hector Levesque, Fiora Pirri, and Ray Reiter. *Foundations for the situation calculus*. 1998.

[146] F Lévy and J Quantz. "Representing beliefs in a situated event calculus". In: *Proceedings of the Thirteenth European Conference on Artificial Intelligence*. Citeseer. 1997.

[147] David Lewis. "Causation". In: *The journal of philosophy* 70.17 (1974), pp. 556–567.

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

[149] S Matthew Liao. "The loop case and Kamm's doctrine of triple effect". In: *Philosophical Studies* 146.2 (2009), p. 223.

[150] S Matthew Liao et al. "Putting the trolley in order: Experimental philosophy and the loop case". In: *Philosophical Psychology* 25.5 (2012), pp. 661–671.

[151] Vladimir Lifschitz. "Action languages, answer sets, and planning". In: *The Logic Programming Paradigm*. Springer, 1999, pp. 357–373.

[152] Vladimir Lifschitz. "What Is Answer Set Programming?." In: *AAAI*. Vol. 8. 2008, pp. 1594–1597.

[153] Vladimir Lifschitz and Hudson Turner. "Representing transition systems by logic programs". In: *Logic Programming and Nonmonotonic Reasoning*. Springer, 1999, pp. 92–106.

[154] Fangzhen Lin and Yuting Zhao. "ASSAT: Computing answer sets of a logic program by SAT solvers". In: *Artificial Intelligence* 157.1-2 (2004), pp. 115–137.

[155] Andrea Loreggia et al. "Preferences and Ethical Principles in Decision Making". In: *Proceedings of the 1st AAAI/ACM Conference on AI, Ethics, and Society (AIES)*. 2018.

[156] Emiliano Lorini. "On the logical foundations of moral agency". In: *International Conference on Deontic Logic in Computer Science*. Springer. 2012, pp. 108–122.

[157] Emiliano Lorini, Dominique Longin, and Eunate Mayor. "A logical analysis of responsibility attribution: emotions, individuals and collectives". In: *Journal of Logic and Computation* 24.6 (2013), pp. 1313–1339.

[158] D Luce and H Raffia. *Games and decisions. Mineola, NY*. 1985.

[159] John Mackie. *Ethics: Inventing right and wrong*. Penguin UK, 1990.

[160] John L Mackie and JL MacKie. *The cement of the universe*. Oxford, 1980.

[161] Bertram F Malle et al. "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*. ACM. 2015, pp. 117–124.

[162] Victor W Marek and Miroslaw Truszczyński. "Stable models and an alternative logic programming paradigm". In: *The Logic Programming Paradigm*. Springer, 1999, pp. 375–398.

[163] David Marr. "A computational investigation into the human representation and processing of visual information". In: *Vision* (1982), pp. 125–126.

[164] P Marti et al. "Engaging with artificial pets". In: *Proceedings of the 2005 annual conference on European association of cognitive ergonomics*. University of Athens. 2005, pp. 99–106.

[165] Norman Clayton McCain. "Causality in commonsense reasoning about actions". PhD thesis. University of Texas at Austin, 1997.

[166] John McCarthy and Patrick J Hayes. "Some philosophical problems from the standpoint of artificial intelligence". In: *Readings in artificial intelligence* (1969), pp. 431–450.

[167] Bruce M McLaren. "Computational models of ethical reasoning: Challenges, initial steps, and future directions". In: *IEEE intelligent systems* 4 (2006), pp. 29–37.

[168] Mario F Mendez, Eric Anderson, and Jill S Shapira. "An investigation of moral judgement in frontotemporal dementia". In: *Cognitive and behavioral neurology* 18.4 (2005), pp. 193–197.

[169] Peter Menzies. "Counterfactual theories of causation". In: (2001).

[170] Simone Migliore et al. "Counterfactual thinking in moral judgment: an experimental study". In: *Frontiers in psychology* 5 (2014).

[171] J Mikhail. "Universal moral grammar: Theory, evidence and the future". In: *Trends in cognitive sciences* 11.4 (2007), pp. 143–152.

[172] John Stuart Mill. "Utilitarianism and other essays". In: (1987).

[173] Rob Miller and Murray Shanahan. "Some alternative formulations of the event calculus". In: *Computational logic: logic programming and beyond*. Springer, 2002, pp. 452–490.

[174] Ryan M Miller, Ivar A Hannikainen, and Fiery A Cushman. "Bad actions or bad outcomes? Differentiating affective contributions to the moral condemnation of harm." In: *Emotion* 14.3 (2014), p. 573.

[175] George Edward Moore and Thomas Baldwin. *Principia ethica*. Cambridge University Press, 1993.

[176] Michael S Moore. *Causation and responsibility: An essay in law, morals, and metaphysics*. Oxford University Press on Demand, 2009.

[177] Michael S Moore. "Patrolling the borders of consequentialist justifications: the scope of agent-relative restrictions". In: *Law and Philosophy* 27.1 (2008), pp. 35–96.

[178] Erik T Mueller. *Commonsense reasoning: an event calculus based approach*. Morgan Kaufmann, 2014.

[179] Erik T Mueller. "Event calculus and temporal action logics compared". In: *Artificial Intelligence* 170.11 (2006), pp. 1017–1029.

[180] J v Neumann. "Zur theorie der gesellschaftsspiele". In: *Mathematische annalen* 100.1 (1928), pp. 295–320.

[181] Ritesh Noothigattu et al. "A voting-based system for ethical decision making". In: *arXiv preprint arXiv:1709.06692* (2017).

[182] Robert Nozick. *Anarchy, state, and utopia*. 1974.

[183] Ruwen Ogien. *Human Kindness and the Smell of Warm Croissants: An Introduction to Ethics*. Columbia University Press, 2015.

[184] Judea Pearl. "Causality: models, reasoning and inference". In: *Econometric Theory* 19.675-685 (2003), p. 46.

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

[186] L M Pereira and A Saptawijaya. "Moral decision making with ACORDA". In: *Short Paper LPAR* 7 (2007).

[187] Luis Moniz Pereira and Ari Saptawijaya. "Counterfactuals, logic programming and agent morality". In: *Applications of Formal Philosophy*. Springer, 2017, pp. 25–53.

[188] John V Petrocelli et al. "Counterfactual potency." In: *Journal of personality and social psychology* 100.1 (2011), p. 30.

[189] David A Pizarro, Eric Uhlmann, and Paul Bloom. "Causal deviance and the attribution of moral responsibility". In: *Journal of Experimental Social Psychology* 39.6 (2003), pp. 653–660.

[190] Reginald E Plato et al. "The dialogues of Plato". In: (1984).

[191] Thomas M Powers. "Prospects for a Kantian machine". In: *IEEE Intelligent Systems* 21.4 (2006), pp. 46–51.

[192] Arthur N Prior. *Past, present and future*. Vol. 154. Clarendon Press Oxford, 1967.

[193] Philip L Quinn. "Divine commands and moral requirements". In: (1978).

[194] James Rachels et al. "Active and passive euthanasia". In: (1975).

[195] Iyad Rahwan. "Society-in-the-loop: programming the algorithmic social contract". In: *Ethics and Information Technology* 20.1 (2018), pp. 5–14.

[196] Anand S Rao and Michael P Georgeff. "Modeling rational agents within a BDI-architecture." In: *KR* 91 (1991), pp. 473–484.

[197] John Rawls. "A theory of justice (cambridge". In: *Mass.: Harvard University* (1971).

[198] Raymond Reiter. "A logic for default reasoning". In: *Artificial intelligence* 13.1-2 (1980), pp. 81–132.

[199] Raymond Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT press, 2001.

[200] Raymond Reiter. "The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression". In: *Artificial intelligence and mathematical theory of computation: papers in honor of John McCarthy* 27 (1991), pp. 359–380.

[201] Samuel C Rickless. "The doctrine of doing and allowing". In: *The Philosophical Review* 106.4 (1997), pp. 555–575.

[202] William David Ross. *The right and the good*. Oxford University Press, 2002.

[203] Paul Rozin, Linda Millman, and Carol Nemeroff. "Operation of the laws of sympathetic magic in disgust and other domains." In: *Journal of personality and social psychology* 50.4 (1986), p. 703.

[204] Massimo Ruffolo et al. "Exploiting ASP for semantic information extraction". In: *In Proceedings ASP05-Answer Set Programming: Advances in Theory and Implementation*. Citeseer. 2005.

[205] Pasquale Rullo, Chiara Cumbo, and Veronica L Policicchio. "Learning rules with negation for text categorization". In: *Proceedings of the 2007 ACM symposium on Applied computing*. ACM. 2007, pp. 409–416.

[206] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*. Vol. 37. University of Cambridge, Department of Engineering Cambridge, England, 1994.

[207] Bertrand Russell. *Religion and science*. 165. Oxford University Press, USA, 1997.

[208] Stuart Jonathan Russell et al. *Artificial intelligence: a modern approach*. Vol. 2. Prentice hall Upper Saddle River, 2003.

[209] Joe Sachs. "Plato: Republic". In: *Newburyport: Focus Publishing* (2007).

[210] Erik Sandewall. *Features and Fluents. The Representation of Knowledge about Dynamical Systems. Volume I*. 1994.

[211] Erik Sandewall. *Filter preferential entailment for the logic of action in almost continuous worlds*. Universitetet i Linköping/Tekniska Högskolan i Linköping. Institutionen för Datavetenskap, 1989.

[212] Leonard J Savage. "The theory of statistical decision". In: *Journal of the American Statistical association* 46.253 (1951), pp. 55–67.

[213] Geoffrey Sayre-McCord. "Coherence and models for moral theorizing". In: *Pacific Philosophical Quarterly* 66.1-2 (1985), pp. 170–190.

[214] Jana Schaich Borg et al. "Consequences, action, and intention as factors in moral judgments: An fMRI investigation". In: *Journal of cognitive neuroscience* 18.5 (2006), pp. 803–817.

[215] Stephan Schiffel and Michael Thielscher. "Reconciling situation calculus and fluent calculus". In: *AAAI*. Vol. 6. 2006, pp. 287–292.

[216] Marc Serramia et al. "Moral Values in Norm Decision Making". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2018, pp. 1294–1302.

[217] Murray Shanahan. "An abductive event calculus planner". In: *The Journal of Logic Programming* 44.1-3 (2000), pp. 207–240.

[218] Murray Shanahan. *Solving the frame problem: a mathematical investigation of the common sense law of inertia*. MIT press, 1997.

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

[220] Henry Sidgwick. *The methods of ethics*. Hackett Publishing, 1907.

[221] Patrik Simons, Ilkka Niemelä, and Timo Soininen. "Extending and implementing the stable model semantics". In: *Artificial Intelligence* 138.1-2 (2002), pp. 181–234.

[222] P Singer. "Ethics and intuitions". In: *The Journal of Ethics* 9.3-4 (2005), pp. 331–352.

[223] Peter Singer. *A companion to ethics*. John Wiley & Sons, 2013.

[224] Brian Skyrms. "Choice and chance: An introduction to inductive logic". In: (1968).

[225] Steven Sloman, Aron K Barbey, and Jared M Hotaling. "A causal model theory of the meaning of cause, enable, and prevent". In: *Cognitive Science* 33.1 (2009), pp. 21–50.

[226] RD Smith and BD Slenning. "Decision analysis: dealing with uncertainty in diagnostic testing". In: *Preventive Veterinary Medicine* 45.1-2 (2000), pp. 139–162.

[227] Robert Sparrow. "Building a better WarBot: Ethical issues in the design of unmanned systems for military applications". In: *Science and Engineering Ethics* 15.2 (2009), pp. 169–187.

[228] Wolfgang Spohn. "Direct and indirect causes". In: *Topoi* 9.2 (1990), pp. 125–145.

[229] Robert C Stalnaker. "A theory of conditionals". In: *Ifs*. Springer, 1968, pp. 41–55.

[230] Karsten J Struhl and Paula S Rothenberg. *Ethics in perspective: a reader*. Random House, 1975.

[231] Philip E Tetlock et al. "People as intuitive prosecutors: The impact of social-control goals on attributions of responsibility". In: *Journal of Experimental Social Psychology* 43.2 (2007), pp. 195–209.

[232] Michael Thielscher. "A unifying action calculus". In: *Artificial Intelligence* 175.1 (2011), pp. 120–141.

[233] Michael Thielscher. "From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem". In: *Artificial intelligence* 111.1-2 (1999), pp. 277–299.

[234] Michael Thielscher. "Introduction to the fluent calculus". In: (1998).

[235] Richmond H Thomason. "Indeterminist time and truth-value gaps1". In: *Theoria* 36.3 (1970), pp. 264–281.

[236] Judith Jarvis Thomson. "Double effect, triple effect and the trolley problem: Squaring the circle in looping cases". In: *Yale Law Journal* 94.6 (1985), pp. 1395–1415.

[237] Judith Jarvis Thomson. "The trolley problem". In: *The Yale Law Journal* 94.6 (1985), pp. 1395–1415.

[238] Mihnea Tufiş and Jean-Gabriel Ganascia. "Grafting norms onto the BDI agent model". In: *A Construction Manual for Robots' Ethical Systems*. Springer, 2015, pp. 119–133.

[239] Piercarlo Valdesolo and David DeSteno. "Manipulations of emotional context shape moral judgment". In: *PSYCHOLOGICAL SCIENCE-CAMBRIDGE-* 17.6 (2006), p. 476.

[240] Kristof Van Belleghem, Marc Denecker, and Danny De Schreye. "On the relation between situation calculus and event calculus". In: *The Journal of Logic Programming* 31.1 (1997), pp. 3–37.

[241] Ibo Van de Poel, Lambèr Royakkers, and Sjoerd D Zwart. *Moral responsibility and the problem of many hands*. Vol. 29. Routledge, 2015.

[242] Johannes A van der Ven and Hans-Georg Ziebertz. *Tensions within and between religions and human rights*. Vol. 2. Brill, 2012.

[243] Abraham Wald. "Statistical decision functions." In: (1950).

[244] Wendell Wallach and Colin Allen. *Moral machines: Teaching robots right from wrong*. Oxford University Press, 2008.

[245] Wendell Wallach, Colin Allen, and Iva Smit. "Machine morality: bottom-up and top-down approaches for modelling human moral faculties". In: *Ai & Society* 22.4 (2008), pp. 565–582.

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

[247] Thalia Wheatley and Jonathan Haidt. "Hypnotic disgust makes moral judgments more severe". In: *Psychological science* 16.10 (2005), pp. 780–784.

[248] Bernard Williams. *Ethics and the Limits of Philosophy*. Routledge, 2006.

[249] James Woodward. *Making things happen: A theory of causal explanation*. Oxford university press, 2005.

[250] Richard W Wright. "Causation, responsibility, risk, probability, naked statistics, and proof: Pruning the bramble bush by clarifying the concepts". In: *Iowa L. Rev.* 73 (1987), p. 1001.

[251] Yueh-Hua Wu and Shou-De Lin. "A Low-Cost Ethics Shaping Approach for Designing Reinforcement Learning Agents". In: *arXiv preprint arXiv:1712.04172* (2017).

[252] Masahiro Yamamoto and Masafumi Hagiwara. "Moral judgment system using evaluation expressions". In: *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*. IEEE. 2014, pp. 1040–1047.

[253] Ro'i Zultan, Tobias Gerstenberg, and David A Lagnado. "Finding fault: causality and counterfactuals in group attributions." In: *Cognition* 125.3 (2012), pp. 429–440.