A Defeasible Logic Implementation of Ethical Reasoning

Louise A. Dennis , Cristina Perea del Olmo
Department of Computer Science, University of Manchester, UK
louise.dennis@manchester.ac.uk, cristina.pereadelolmo@student.manchester.ac.uk

Abstract

We present an implementation of ethical reasoning using defeasible logic. We evaluate this on two case studies illustrating how changing information can affect the outcome of the ethical reasoning.

We discuss the system's strengths and weaknesses in terms of formal ethical reasoning, concluding that defeasible logic is a promising approach with particular value in the way it separates rules from exceptions but that additional work is needed, particularly in understanding how defeasible reasoning interacts with sequences/plans of action.

1 Introduction

Defeasible logic is a form of non-monotonic reasoning. It has often been used in artificial intelligence for its efficient implementation of rules in a non-monotonic environment.

Defeasible logic divides rules into "strict rules", which are always going to be applied, "defeasible rules", and "defeaters", which represent the rules that might apply.

We present here an implementation of ethical reasoning using defeasible logic and evaluate it on two case studies.

2 Background

2.1 Machine Ethics

Machine ethics is the study of how to implement ethical reasoning in machines. There are a number of approaches to machine ethics, including approaches from symbolic artificial intelligence, which generally take a philosophical theory and operationalize it, and approaches from machine learning which attempt to learn ethical behaviour from observation. Following (Wallach and Allen 2008), symbolic approaches are generally classed as top-down and contrasted to machine learning approaches which are considered bottom-up. There are several approaches that seek to combine these, for instance those in which philosophical theory provides an overarching framework within which details can be established via learning (e.g., (Anderson and Leigh Anderson 2014)).

Popular philosophical theories for the implementation of top-down machine ethics include utilitarianism (in which the outcomes of actions are scored and the action with the highest score is chosen), deontic logic (in which ethics are encoded as rules that explicitly refer to actions that are permitted, obliged or prohibited in specific situations) (Gabbay et al. 2013), and variants on virtue ethics which refer to the extent to which an action is in line with some set of desirable values. Many approaches combine aspects of several philosophical theories – for instance approaches which evaluate the outcomes of actions with respect to values or *ethical principles* and then use rules to select the preferred choice (Anderson and Leigh Anderson 2014).

In (Horty 2012), Horty suggests the use of defeasible logic as an approach to the implementation of deontic logic style machine ethics. In particular, he suggests that "simple oughts" (e.g., "You ought to get vaccinated") should be treated as defeasible statements. So, in general, the simple ought defines the ethical course of action, but there may be exceptions in which the obligation no longer holds. In this paper, we integrate this reasoning over oughts with an existing, easily implementable, defeasible logic.

2.2 Defeasible Logic

Defeasible Logic (Horty 2012; Koons 2017) is a non-monotonic logic that models reasoning in which some conclusions are drawn only in the absence of evidence to the contrary. In this paper, we use the defeasible logic, DL, from (Antoniou et al. 2000) which can be easily translated into Prolog.

Definition 2.1. A *defeasible theory* consists of two sets and a relation; $(F, R, <)$. F is a set of facts in some language, $\mathcal{L}$. R is a set of rules and "$<$", is a superiority relation on the set of rules.

R is divided into three sets:

Strict rules are indisputable. Once the rule antecedant is satisfied, the conclusion always follows. There is no exception to this sort of rule. An example of a strict rule is "whales are mammals".

Defeasible rules represent deductions that usually, but not universally, hold. The conclusions reached following these rules can be retracted if further information is provided. An example of a defeasible rule is "mammals generally live on dry land".

Defeaters represent additional evidence that can prevent a conclusion from being drawn but cannot usually be used themselves to get to a conclusion. They provide adverse information to defeat a defeasible rule. An example of a defeater is "whales do not live on dry land".

The superiority relation "$<$" on R is anti-symmetic and transitive and so defines a partial order on R. For any two rules, r_1, r_2, if $r_1 < r_2$, we give preference to r_2 over r_1 when drawing a conclusion.

Example 1. Let us say Charly is a whale. Then "Charly is a whale" $\in F$ ("Charly is a whale" is a fact). Let the rule "whales are mammals" be a strict rule, "mammals generally live on dry land" is a defeasible rule (denoted r_1), and "whales do not live on dry land" is a defeater (denoted r_2). Let $r_1 < r_2$.

Since Charly is a whale, it is always a mammal. Mammals generally live on dry land, so Charly should live on dry land, but since $r_1 < r_2$, we cannot conclude that Charly inhabits dry land. The following figure based on *The Tweety Triangle* (Horty 2012) illustrates the reasoning we follow that prevents us from concluding that Charly lives on dry land.

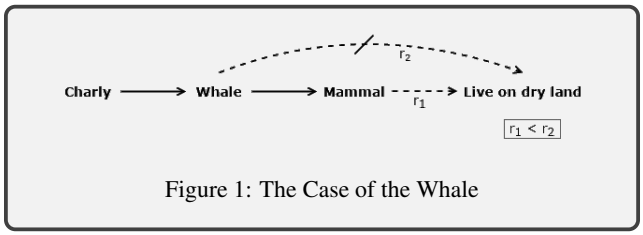

Figure 1: The Case of the Whale

Note that if we had wanted to be able to conclude that Charly does not live on dry land, then we would have had to represent "whales do not live on dry land" as a defeasible rule, not as a defeater. As it is, we can only fail to conclude that Charly lives on dry land.

3 Extending DL with Oughts

We will work with a term language, $\mathcal{L}$, constructed from variables, constants and functions in the normal way. A *literal* is a proposition p or its negation $\neg p$. The complement of a literal is denoted $\sim p$, i.e. if p is positive its complement $\sim p$ is $\neg p$, if p is negative ($\neg p$) then its complement $\sim p$ is p.

Following Horty, we extend this language with the "ought" operator, $\bigcirc$, such that if l is a literal and c a constant in $\mathcal{L}$ then $\bigcirc(c, l)$ is interpreted as "c ought to do l". We will refer to these and their complements, as *ought* statements[1].

As well as concerning ourselves with the complements of ought statements (e.g., "c ought to do l" and "it is not the case that c ought to do l"), we are also interested in reasoning about the complements of the literals contained within the ought statements (e.g., "c ought not to do l"). We will therefore also introduce the concept of an *ought complement*, $\sim_o$.

Definition 3.1. If p is an ought statement (either $\bigcirc(c, l)$ or $\neg \bigcirc (c, l)$) then $\sim_o p$ (the *ought complement*) is $\bigcirc(c, \sim l)$ or $\neg \bigcirc (c, \sim l)$ respectively.

[1] As pointed out by one of the referees, the ought operator functions as the obligation operator in most deontic logics. We preserve Horty's terminology here but agree that obligation is an equivalent term.

The following formalisation of defeasible logic is the formalisation of DL from (Antoniou et al. 2000) extended to handle ought statements as defeasible rules.

Definition 3.2. A *rule* is a triple, $\langle r, A, B\rangle$ which we will write $r : A \rightarrow B$ in which r is a label, A (the antecedants) is a finite set of literals and ought statements and B (the consequent) is a literal or ought statement.

We will write $A(r)$ to indicate the antecedants of the rule labelled, r, and $B(r)$ for the consequent of the rule labelled, r.

Let D be the defeasible theory $(F, R, <)$. A conclusion to the defeasible theory D is a literal or ought statement that has a tag Δ or ∂. It can have one of the following four forms:

(i) $+\Delta q$: q is definitely provable (i.e. can be proved) in D.

(ii) $-\Delta q$: it has been proved that q is not definitely provable in D.

(iii) $+\partial q$: q is defeasibly provable in D.

(iv) $-\partial q$: it has been proved that q is not defeasibly provable in D.

The rules R of D are divided into three distinct sets: the strict rules R_s, the defeasible rules R_d and the defeaters R_{df}. We write R_{sd} for $R_s \cup R_d$. $R[q]$ denotes the set of all rules in R with conclusion unifiable with q. All rules containing ought statements as their consequent are defeasible. Without loss of generality we treat "simple oughts" as rules with empty antecedants (i.e., $[] \rightarrow \bigcirc(c, l)$).

Definition 3.3. A derivation is a finite sequence $P = P(1), \ldots, P(n)$ of tagged literals and ought statements. The initial section of a sequence P of length i is denoted $P(1..i)$. A derivation obeys the following conditions:

1. if $P(i+1) = +\Delta q$ then either
   (a) $q \in F$ or
   (b) $\exists r \in R_s[q]. \forall a \in A(r). + \Delta a \in P(1..i)$
   A literal is definitely provable if it is a fact, or derived using only strict rules from facts.
2. if $P(i+1) = -\Delta q$ then both
   (a) $q \notin F$ and
   (b) $\forall r \in R_s[q]. \exists a \in A(r). - \Delta a \in P(1..i)$
   A literal is not definitely provable if it is not a fact and can not be derived using only strict rules from facts.
3. if $P(i+1) = +\partial q$ then either
   (a) $+\Delta q \in P(1..i)$ or
   (b) i. $\exists r \in R_{sd}[1]. \forall a \in A(r). + \partial a \in P(1..i)$ and
   ii. $-\Delta \sim q \in P(1..i)$ and
   iii. $\forall s \in R[\sim q]$ either
   A. $\exists a \in A(s). - \partial a \in P(1..i)$ or
   B. $\exists t \in R_{sd}[q]$ such that $\forall a \in A(t). + \partial a \in P(1..i)$ and $t > s$
   iv. $\forall s \in R[\sim_o q]$ either
   A. $\exists a \in A(s). - \partial a \in P(1..i)$ or
   B. $\exists t \in R_{sd}[q]$ such that $\forall a \in A(t). + \partial a \in P(1..i)$ and $t > s$

A literal or ought statement, q, is defeasibly provable if it is definitely provable or it can be derived from other defeasibly provable statements and its complement can not be definitely proved and if its complement can be derived in R then any rule (including defeaters) that can be used to deduce the complement (and ought complement where relevant) has lower precedence than a rule that can be used to deduce q.

4. if $P(i+1) = -\partial q$ then both
   (a) $-\Delta q \in P(1..i)$ and
   (b) i. $\forall r \in R_{sd}[1]. \exists a \in A(r). -\partial a \in P(1..i)$ or
   ii. $+\Delta \sim q \in P(1..i)$ or
   iii. $\exists s \in R[\sim q]$ such that
      A. $\forall a \in A(s). +\partial a \in P(1..i)$ and
      B. $\forall t \in R_{sd}[q]$ either $\exists a \in A(t). -\partial a \in P(1..i)$ or $t \not> s$
   iv. $\exists s \in R[\sim_o q]$ such that
      A. $\forall a \in A(s). +\partial a \in P(1..i)$ and
      B. $\forall t \in R_{sd}[q]$ either $\exists a \in A(t). -\partial a \in P(1..i)$ or $t \not> s$

   A literal or ought statement, q, is not defeasibly provable if it is not definitely provable, and if it can be derived from other defeasibly provable statements, then either its complement can be definitely proved or if its complement can be derived using some rule (possibly a defeater) that has higher precedence than any rule that can be used to deduce q.

Note this definition differs from that in (Antoniou et al. 2000) only by the inclusion of ought statements in rules and by 3.b).iv and 4.b).iv. These are extensions to the original proof theory which specifically handle ought complements.

Informally, the use of ought complements means that we can not deduce both $\bigcirc(c, l)$ and $\bigcirc(c, \sim l)$ (c is both obliged to do l and obliged not to do l) since we can not conclude $\bigcirc(c, l)$ if $\bigcirc(c, \sim l)$ is also derivable from the rules unless there is a rule for deriving $\bigcirc(c, l)$ that has higher precedence than any rule that derives $\bigcirc(c, \sim l)$. However, we leave a formal proof of this for further work.

3.1 Implementation

An advantage of using DL as our defeasible logic is that it has a natural translation into Prolog. For instance, we can represent the conditions for definitely provable in Prolog as:

```
provable(definite, C, [C]):- fact(C).
```

If `C` is a fact then it can be derived with derivation `[C]`.

```
provable(definite, C, [C | [R | D]]):-
    strict(R, A, C),
    provable_list(definite, A, D).
```

If `C` is the consequent of a strict rule, `R`, and its antecedants, `A`, can all be definitely proved with derivation `D` – then `C` can be definitely proved with derivation `[C | [ R | D]]` (we use the Prolog notation here where `[H | T]` denotes a list with head element `H` and tail element a list `T`). We include the rules in the derivations we output, even though this does not conform to the definition in 3.3 since this assists with understanding proofs and debugging.

The following code proves all statements in a list (i.e., all the antecedants of a rule) and combines the derivations.

```
provable_list(_, [], []).
provable_list(Type, [H|T], Out):-
    provable(Type, H, Out1),
    provable_list(Type, T, Out2),
    append(Out1, Out2, Out).
```

We are able to leverage Prolog's built-in support for unification to handle the instantiation of variables in literals and ought statements and back-tracking search to allow us to search over multiple possible rules.

Space prevents us including the full implementation here – for instance Prolog code for handling defeasible rules and defeaters. The implementation follows the semantics from Section 3 and can be found together with our case studies, at `https://github.com/louiseadennis/defeasible`.

4 Case Studies

4.1 Covid-19 Rules and Guidance

We use the rules and guidance around Covid-19 infection control as a simple illustration of our implementation. Consider the following set of rules where we follow the Prolog convention and represent variables as upper case letters:

$$
\begin{array}{ll}
o_1: & [] \rightarrow \bigcirc(X, get\_vaccinated) \\
o_2: & [] \rightarrow \bigcirc(X, facemask) \\
d_1: & [health\_c(X)] \rightarrow \neg(\bigcirc(X, get\_vaccinated)) \\
d_2: & [facemask\_e(X)] \rightarrow \neg(\bigcirc(X, facemask)) \\
d_3: & [\neg(vaccinated(X))] \rightarrow \bigcirc(X, \neg(go\_to\_work)) \\
d_4: & [\neg(vaccinated(X))] \rightarrow \bigcirc(X, \neg(go\_to\_pub)) \\
df_1: & [keyworker(X)] \rightarrow \neg \bigcirc (X, \neg(go\_to\_work))
\end{array}
$$

where

$$
\begin{array}{c}
d_1 > o_1 \\
d_2 > o_2 \\
df_1 > d_3
\end{array}
$$

o_1, o_2, d_1, d_2, d_3 and d_4 are defeasible rules. df_1 is a defeater. These rules state that a person ought to get vaccinated and wear a facemask. However if they have a health contraindication ($health\_c$) they are not obliged to get vaccinated and if they have a facemask exemption ($facemask\_e$) they are not obliged to wear a facemask. A person who is not vaccinated ought not go to work or to the pub. However, a keyworker is not obliged not to go to work.

Now let us consider two individuals: Ms C. and Ms L. Neither Ms C. nor Ms L. are vaccinated. In addition, Ms C. is a key worker and has a health contraindication for the vaccine.

$$
\begin{array}{l}
health\_c(msc) \\
keyworker(msc) \\
\neg vaccinated(msc) \\
\neg vaccinated(msl)
\end{array}
$$

Prolog allows us to place a variable in our queries so we can ask:

```
provable(defeasible, ought(msc, X), D)
```

to get something, X, that Ms C. (resp. Ms L.) ought to do (and the derivations that lead to those conclusions). Furthermore, Prolog allows us to reject answers to get an alternative instantiation for X so we can generate a list of all the things that ought to be done.

This process tells us that Ms C. ought not to go to the pub and that she ought to wear a facemask. Ms L., additionally, ought to get vaccinated and ought not to go to work.

A key advantage of the defeasible logic representation here is that we can separate out the exceptions to rules from the rule that should generally be followed. We would argue that this makes is easier for stake-holders to understand the formalisation since it avoids the need for rules which have a long list of antecedents excluding exceptions to the rule. It also has the advantage that in any iterative process of formalisation, exceptions can be added easily as they are developed without needing to alter pre-existing rules. This has particular value in situations which may be rapidly evolving.

4.2 Treatment Refusal

Our second case study is based on a discussion of medical ethics in (Edwards, McCarthy, and Konishi 2010).

To proceed with any medical procedure, all the way from a simple x-ray to an organ transplant, a patient must give their consent. This is because every person has the right to autonomy (Miller 1981).

Mrs A. is an 88-year-old woman who lives alone. Her daughter is the only family she has and lives in the same city. Mrs A. was hospitalized with anemia, lower leg edema and ascities. Her hemoglobin levels had decreased from 12 to 4 in six months. The anemia might have been caused by stomach bleeding; if this were the case, the consequences of not proceeding with further treatment would be fatal. To proceed with treatment an examination is needed, starting with a gastroscopy.

After hearing all this information, Mrs A. refuses to proceed with any further treatment or examination.

As noted, the patient's decision should be respected, nevertheless, there are other factors to consider. In particular, is Mrs A. capable of making a decision? For instance, the low levels of hemoglobin associated with anemia can affect decision making.

Even where a lack of capacity can be established, it is important to bear in mind the possible existence of a *living will* and the opinion of the next of kin (NHS). A living will is a formal document in which a person specifies a decision taken in advance regarding some sort of treatment (more typically a refusal of this treatment). This lets the professionals treating the patient know what their decision was when they were in a more optimal situation. If a patient is not capable of making a decision then their next of kin should be consulted.

We formalised the rules for this situation as follows:

$$
\begin{aligned}
d_1 &: [dec\_hem\_lev\_low(X)] \rightarrow anemia(X)\\
d_2 &: [anemia(X)] \rightarrow stomach\_bleeding(X)\\
d_3 &: [stomach\_bleeding(X)] \rightarrow death(X)\\
d_4 &: [\bigcirc(doctor, do(treat, X))] \rightarrow \bigcirc(doctor, do(examine, X))\\
d_5 &: [death(X)] \rightarrow \bigcirc(doctor, do(treat, X))\\
d_6 &: [\neg(right\_mind(X))] \rightarrow \neg(refuse\_consent(X, Action))\\
d_7 &: [anemia(X)] \rightarrow \neg(right\_mind(X))\\
d_8 &: [living\_will(X)] \rightarrow right\_mind(X)\\
d_9 &: [\neg(right\_mind(X))] \rightarrow \bigcirc(doctor, consult\_nok(X))\\
d_{10} &: [\bigcirc(X, do(examine, Y))] \rightarrow \bigcirc(X, obtain\_consent(Y))\\
d_{11} &: [\bigcirc(X, do(treat, Y))] \rightarrow \bigcirc(X, obtain\_consent(Y))\\
df_1 &: [refuse\_consent(X, A)] \rightarrow \bigcirc(doctor, not(do(A, X)))
\end{aligned}
$$

where:

$$
\begin{aligned}
d_4 &< df_1\\
d_5 &< df_1\\
d_{10} &< df_1\\
d_{11} &< df_1\\
d_9 &< df_1\\
d_7 &< d_8\\
df_1 &< d_6
\end{aligned}
$$

Rules $d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}$ and d_{11} are defeasible and rule df_1 is a defeater.

We examined three variations on the case with this encoding. In the first variation, Mrs A. has been diagnosed with decreased hemoglobin and anemia.

$$
\begin{aligned}
&dec\_hem\_lev\_low(mrsa)\\
&anemia(mrs)
\end{aligned}
$$

The system recommends that the doctor ought to obtain consent, perform an examination, treat the patient and consult their next of kin.

In the second variation, Mrs A. has refused consent.

$$
\begin{aligned}
&dec\_hem\_lev\_low(mrsa)\\
&anemia(mrs)\\
&refuse\_consent(mrsa, treat)
\end{aligned}
$$

In this case, the system continues to recommend that the doctor obtains consent, performs an examination, treats the patient and consults their next of kin. Essentially the doubts around Mrs A.'s capacity arising from her anemia mean that in both cases (whether she has or has not expressed her refusal of consent), the next of kin should be consulted.

In the final variation, Mrs A. has refused consent via a living will.

$$
\begin{aligned}
&dec\_hem\_lev\_low(mrsa)\\
&anemia(mrs)\\
&refuse\_consent(mrsa, treat)\\
&living\_will(mrsa)
\end{aligned}
$$

In this case the system does not recommend that the doctor ought to do anything.

Once again the formalisation benefits from our ability to separate general rules and their exceptions so that no rule becomes too involved. However there were a number of challenges faced in creating the formalisation and a number of areas in which it is not quite satisfactory which we will discuss further in section 6.

5 Related Work

Taking what is essentially a logic programming based approach to machine ethics is not new.

The Deontic Cognitive Event Calculus (Bringsjord, Arkoudas, and Bello 2008) uses automated reasoning with an explicit logical formalism that incorporates both deontic modalities, such as the $\bigcirc$ modality we use here and the Event Calculus (Miller and Shanahan 2002) for reasoning about the outcomes of actions. (Berreby, Bourgne, and Ganascia 2017) presents a framework based on answer set programming, which is more powerful than the Prolog approach taken here that can be used to model reasoning in a variety of ethical theories. There has also been work on formalising ethical reasoning in Belief-Desire-Intention programming languages (Dennis et al. 2016; Bremner et al. 2019; Cardoso et al. 2021) which have many logic programming features.

Our work here aims to ask what defeasible logic can add to this field rather than establishing that a logic programming approach to machine ethics is feasible. Moreover, we believe a key contribution is our attempt to formalise two case studies within this framework and so highlight both some of its strengths and weaknesses.

6 Discussion

We have used an implemented defeasible logic to explore the utility of such logics for machine ethics. The implementation has a number of positive features – in particular, and unsurprisingly, the ability to state rules without having to list all possible exceptions makes the formalisation clearer in many places than it might otherwise be. However, there are a number of areas where the formalism remains not entirely satisfactory. In particular, there are a number of issues around time, sequencing and planning that the system is ill-equipped to handle.

In the case of Mrs A, we saw that where we had simply the medical facts of the case and no information about consent, the system recommended both that an examination be performed and, because a doctor ought to obtain consent before performing an examination, that the doctor obtain consent. However, the recommendation did not capture the fact that obtaining consent should occur before the examination, nor that the obligation to perform the examination would be contingent upon the result of the attempt to obtain consent[2]. We could have formalised this differently – for instance rule d_4 could have explicitly noted the need for consent:

$$[stomach\_bleeding(X), given\_consent(X)] \rightarrow \bigcirc(doctor, do(examine(X)) \quad (1)$$

But then we would have needed a separate rule to say that if a patient may have stomach bleeding then consent for an examination should be obtained and, again, this does not seem to quite capture the medical reasoning involved in this situation where it is the obligation to perform an examination that generates the obligation to obtain consent first[3].

Related, but different, issues arose when we attempted to formalise arguments in favour of and against vaccine passports. For instance, a government should seek to enable international travel:

$$[] \rightarrow \bigcirc(government, enable\_international\_travel)$$

and that vaccine passports would assist this:

$$[imp\_vaccine\_passport] \rightarrow enable\_internal\_travel$$

However, as implemented, our logic is unable to combine these two rules to infer that the government should implement vaccine passports. It seemed artificial to formalize this second rule as one that said if a government ought to enable travel, then it has an obligation to implement vaccine passports[4]. In deontic logic, the maxim "ought implies can" is well established – i.e., one is only obliged to perform actions one is able to perform. Ideally, when encountering an obligation, our system would then also seek to understand what conditions were needed to make it possible to fulfill that obligation and so generate additional obligations based on making the ought possible. This suggests, at the very least, that the semantics of the defeasible logic should allows us to infer $\bigcirc(P, X)$ if $X \rightarrow Y$ and $\bigcirc(P, Y)$ which the existing semantics does not allow.

More generally, we would like a system that can reason about outcomes of actions and infer obligations to some plan of action. This might involve some kind of combination with a logic of action (Segerberg, Meyer, and Kracht 2020), and/or consideration of the ethics of plans (Lindner, Mattmüller, and Nebel 2020).

A Defeasible logic approach to machine ethics also suffers, as do all top-down approaches to machine ethics from the ethical knowledge engineering problem – where the outcomes of reasoning are only ethical if the rules and priorities are correctly specified.

The Geneth system (Anderson and Leigh Anderson 2014) proposes that the behaviour of autonomous ethical agents

[2] Although superficially similar, the same is not true of the sequencing of the treatment and the examination since the examination is part of the treatment.

[3] As observed by one of the referees, this isn't necessarily a deficiency in the existing semantics since we could have chosen to formalise the example in a way that represented time points, and their sequencing, in the arguments to predicates. The risk here, of course, is that the representation becomes more complex and harder for stake-holders to follow. The question of at what point a formalisation becomes "too involved" is complex and probably, itself, hard, if not impossible, to formalise.

[4] Obviously, our attempt at a formalization also captured similar obligations that would generate an obligation *not* to implement vaccine passports with the precedence relation used to decide among these.

should be guided by the consensus of ethicists. To do this, they create an ethical dilemma analyser that uses concepts from machine learning to identify and encode a consensus opinion in the form of priorities between ethical principles. A similar approach could be applied to the learning of precedences (and potentially defeaters) in a defeasible logic implementation of machine ethics. Such an approach might also need to be combined with techniques that allow users (or groups of appropriate stakeholders) to alter precedences and rules in some principled way after deployment (e.g., as discussed in (Dennis et al. 2021)).

Nevertheless, defeasible logic is clearly a promising approach to the implementation of machine ethics, at least in some application domains, and is worth further investigation.

Acknowledgements

This work was funded by EPSRC Grant EP/V026801/1 *Trustworthy Autonomous Systems Verifiability Node*. Thanks are due to John F. Horty and Schloß Dagstuhl – Leibniz-Zentrum für Informatik for informative discussions that gave rise to the ideas in this paper during Dagstuhl Seminar 19171: *Ethics and Trust: Principles, Verification and Validation*.

Thanks are due to the anonymous referees for raising some thoughtful discussion points.

References

Anderson, M., and Leigh Anderson, S. 2014. GenEth: A General Ethical Dilemma Analyzer. In *Proc. AAAI-14*.

Antoniou, G.; Billington, D.; Governatori, G.; Maher, M. J.; and Rock, A. 2000. A family of defeasible reasoning logics and its implementation.

Berreby, F.; Bourgne, G.; and Ganascia, J.-G. 2017. A declarative modular framework for representing and applying ethical principles. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, 96–104. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Bremner, P.; Dennis, L. A.; Fisher, M.; and Winfield, A. F. 2019. On proactive, transparent and verifiable ethical reasoning for robots. *Proceedings of the IEEE special issue on Machine Ethics: The Design and Governance of Ethical AI and Autonomous Systems* 107:541–561.

Bringsjord, S.; Arkoudas, K.; and Bello, P. 2008. Toward a General Logicist Methodology for Engineering Ethically Correct Robots. *IEEE Intelligent Systems* 21(4):38–44.

Cardoso, R. C.; Ferrando, A.; Dennis, L.; and Fisher, M. 2021. Implementing ethical governors in bdi. In *Proceedings of the 9th International Workshop in Engineering Multi-Agent Systems*. To appear.

Dennis, L.; Fisher, M.; Slavkovik, M.; and Webster, M. 2016. Formal verification of ethical choices in autonomous systems. *Robotics and Autonomous Systems* 77:1–14.

Dennis, L. A.; Bentzen, M. M.; Lindner, F.; and Fisher, M. 2021. Verifiable machine ethics in changing contexts. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 11470–11478. AAAI Press.

Edwards, S.; McCarthy, J.; and Konishi, E. 2010. Case study. Nursing ethics of treatment refusal by patients in Japan. *Nursing ethics* 17:523–6.

Gabbay, D.; Horty, J.; Parent, X.; van der Meyden, R.; and van der Torre, L., eds. 2013. *Handbook of Deontic Logic and Normative Systems.* College Publications, London, UK.

Horty, J. F. 2012. *Reasons as defaults*. Oxford University Press.

Koons, R. 2017. Defeasible Reasoning. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2017 edition.

Lindner, F.; Mattmüller, R.; and Nebel, B. 2020. Evaluation of the moral permissibility of action plans. *Artificial Intelligence* 287:103350.

Miller, R., and Shanahan, M. 2002. *Some Alternative Formulations of the Event Calculus*. Berlin, Heidelberg: Springer Berlin Heidelberg. 452–490.

Miller, B. L. 1981. Autonomy & the refusal of lifesaving treatment. *Hastings Center Report* 22–28.

NHS. Do I have the right to refuse treatment? https://www.nhs.uk/common-health-questions/nhs-services-and-treatments/do-i-have-the-right-to-refuse-treatment.

Segerberg, K.; Meyer, J.-J.; and Kracht, M. 2020. The Logic of Action. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2020 edition.

Wallach, W., and Allen, C. 2008. *Moral Machines: Teaching Robots Right from Wrong*. USA: Oxford University Press, Inc.