Toward the Engineering of Virtuous Machines

Naveen Govindarajulu
Selmer Bringsjord and Rikhiya Ghosh
Rensselaer AI & Reasoning Lab
Rensselaer Polytechnic Institute, (RPI)
Troy, New York 12180, USA
www.rpi.edu

Vasanth Sarathy
Human Robot Interaction Laboratory
Tufts University
Medford, MA 02155, USA
www.tufts.edu

Abstract
While various traditions under the ‘virtue ethics’ umbrella have been studied extensively and advocated by ethicists, it has not been clear that there exists a version of virtue ethics rigorous enough to be a target for machine ethics (which we take to include the engineering of an ethical sensibility in a machine or robot itself, not only the study of ethics in the humans who might create artificial agents). We begin to address this by presenting an embryonic formalization of a key part of any virtue-ethics theory: namely, the learning of virtue by a focus on exemplars of moral virtue. Our work is based in part on a computational formal logic previously used to formally model other ethical theories and principles therein, and to implement these models in artificial agents.

Introduction
What is virtue ethics? One way of summarizing virtue ethics is to contrast it with the two main families of ethical theories: deontological ethics (D) and consequentialism (C). Ethical theories in the family C that are utilitarian in nature hold that actions are morally evaluated based on their total utility (or total disutility) to everyone involved. The best action is the action that has the highest total utility. In stark contrast, ethical theories in D emphasize inviolable principles, and reasoning from those principles to whether actions are obligatory, permissible, neutral, etc. In a departure from both D and C, ethical theories in the virtue-ethics family V are overall distinguished by the principle that the best action in a situation, morally speaking, is the one that a virtuous person would perform (Annas 2011). A virtuous person is defined as a person who has learned and internalized a set of habits or traits termed virtuous. For a virtuous person, virtuous acts become second-nature, and hence are performed in many different situations, through time.

While there has been extensive formal and rigorous modeling done in D and C, there has been little such work devoted to formalizing and mechanizing V. Note that unlike D and C, it is not entirely straightforward how one could translate the concepts and principles in V into a form that is precise enough to be realized in machines. Proponents of V might claim that it is not feasible to do so given V’s emphasis on persons and traits, rather than individual actions or consequences. From the perspective of machine ethics, this is not satisfactory. If V is to be on equal footing with D and C for the purpose of building morally competent machines, AI researchers need to start formalizing parts of virtue ethics, and to then implement such formalization in computation.

We present one such formalization herein; one that uses learning and is based on a virtue-ethics theory presented by Zagzebski (Zagzebski 2010). The formalization is presented courtesy of an expressive computational logic that has been used to model principles in both C and D (e.g. (Govindaraju and Bringsjord 2017a; Govindarajulu et al. 2017)). The formalization answers, abstractly, the following two questions:

<table>
<thead>
<tr>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q1)</td>
</tr>
<tr>
<td>(Q2)</td>
</tr>
</tbody>
</table>

The plan for the paper is as follows. First, we briefly look at why virtuous machines might be useful, and then we briefly cover related work that can be considered as formalizations of virtue ethics. Next, we present an overview of virtue ethics itself, and specifically show that an emphasis on moral exemplars makes good sense for any attempt to engineer a virtuous machine. We next present one version of virtue ethics, VZ (Zagzebski’s version of virtue ethics), that we seek to formalize fully. Then, our calculus and the formalization itself (VZ) are presented. We conclude by discussing future work and remaining challenges.

Why Virtuous Robots?
Note that we do not advocate that machine ethicists pursue virtue ethics over other families of ethical theories. Our goal in the present paper is merely to formalize one version of virtue ethics within the family V. That said, why might virtue ethics be considered over consequentialism or deontological ethics for building morally competent machines? To partially answer this question, we take a short digression into a series of conditions laid out by Alfano, and characterized as identifying the core of virtue ethics:

1Both the families C and D are crisply explained as being in conformity with what we say here in e.g. (Feldman 1978).

2See (Bringsjord, Arkoudas, and Bello 2006) for an introduction to the logicist methodology for building ethical machines.
shown that Hurka’s account is different from rigorous and virtue (Miles 2013). More problematic is that it must be that Hurka’s definition might not capture central aspects of virtues and vices, and that Hurka’s account are rectifiable (Hiller 2011), we feel despite this, and despite our sense that the main problems of maximizing happiness, and are forbidden exactly when they fail to so maximize. A parallel state-of-affairs holds for at least basic knowledge of deontological ethical theories (= family \( \mathcal{D} \)): most readers have for instance some familiarity with Kant’s moral system in family \( \mathcal{D} \), and specifically with his famous “categorical imperative,” which, paraphrasing, says that, unconditionally, one must always act in such a way that this behavior could be universalized.\(^3\) In addition, generally people are familiar with the core tenet of divine-command ethical theories (i.e. of members of \(\mathcal{D} \)), which is (approximately) that actions are obligatory for humans if and only if God commands that these actions be performed (a particular member of \(\mathcal{D} \) is specified in (Quinn 1978)). However, in our experience the epistemic situation is radically different when it comes to the family of ethical theories \( \text{virtue ethics} = (\mathcal{V}) \). For while it’s true that generally educated people can be assumed to be acquainted with the concept of virtue, and with many things long deemed to be virtues (e.g. bravery), an understanding of virtue ethics at the level of ethical \( \text{theory} \) cannot be assumed. We therefore now provide a rapid (and admittedly cursory) synopsis of \( \mathcal{V} \), by drawing from (Vallor 2016), and to some degree from (Annas 2011). It will be seen that \( \mathcal{V} \) makes central use of exemplars, and of learning and development that revolves around them. Hence we shall arrive at a convenient entry point for our AI work devoted to trying to design and build a virtuous machine.

Obviously we cannot in the span of the space we have at hand do full justice to the book-length treatment of \( \mathcal{V} \) that is (Vallor 2016). But we can quickly establish that our technical work, in its focus on the cultivation of virtue for a machine via learning from exemplars, is not merely based on a single, idiosyncratic member of \( \mathcal{V} \), and on one peripheral aspect of this member. On the contrary, study of the work of Vallor and other scholars concerned with a characterization of the family \( \mathcal{V} \) confirms that our exploitation specifically

\[^3\text{This imperative is first set out in} — \text{as it’s known in abbreviation — \textit{Groundwork}; see (Kant 19971785). It’s generally thought by ethicists, and this may be convenient for machine/AI ethics, that Kant had in mind essentially a decision procedure to follow in the attempt to behave in an ethically correct manner. For a lucid and laconic overview of this point, see (Johnson 20042016); and cf. (Powers 2006).} \]
of Zagzebski’s (Zagzebski 2010) focus, from the standpoint of the field of ethics itself, is a worthy point of entry for AI researchers.

To begin, Vallor, drawing on and slightly adapting Van Norden’s (Van Norden 2007) sets out a quartet of commonalities that at least seem to be true of all members of $V$, and the second one is: “A conception of moral virtues as cultivated states of character, manifested by those exemplary persons who have come closest to achieving the highest human good” (¶5, §2.2). But given our specific efforts toward engineering a virtuous machine, it is important to note that Vallor specifically informs us about the key concepts of exemplars in the particular members of the $V$ family; to pick just one of many available places, she writes:

Buddhism’s resonances with other classical virtue traditions do not end here. As with the central role granted by Confucian and Aristotelian ethics to ‘exemplary persons’ (the junzi and phronimoi respectively), bodhisattvas (persons actively seeking enlightenment) generally receive direction to or assistance on the path of self-cultivation from the community of exemplary persons to which they have access. In Buddhism this is the monastic community and lay members of the Sangha . . .[¶5, §2.1.3, (Vallor 2016)]

We said above that we would also draw, albeit briefly, from a second treatment of $V$, viz. (Annas 2011), in order to pave the way into our AI-specific, exemplar-based technical work. About this second treatment we report only that it is one based squarely on a “range of development” (¶3, §Right Action in Ch. 3), where the agent (a human in her case) gradually develops into a truly virtuous person, beginning with unreflective adoption of direct instruction, through a final phase in which “actions are based on understanding gained through experience and reflection” (ibid.). Moreover, Annas explicitly welcomes the analogy between an agent’s becoming virtuous, and an agent’s becoming, say, an excellent tennis-player or pianist. The idea behind the similarity is that “two things are united: the need to learn and the drive to aspire” (emphasis hers: ¶4 Ch. 3). In addition, following Aristotle on $V$ (e.g. see (Aristotle 2000) 1103), no one can become a master tennis-player or pianist without, specifically, playing tennis/the piano with an eye to the mastery of great exemplars in these two domains.

In order to now turn to specific AI work devoted to engineering a virtuous machine, we move from completed consideration of the general foundation of $V$, and its now-conformed essential use of moral exemplars, to a specific use of such exemplars that appears ripe for mechanization.

**Exemplarist Virtue Theory**

Exemplarist virtue theory ($V_e$) builds on the direct reference theory (DRT) of semantics. Briefly, in DRT, given a word or term $w$, its meaning $\mu(w)$ is determined by what the word points out, say $p$, and not by some definition $d$. For example, for a person to use the word “water,” in a correct manner, that person neither needs to possess a definition of water nor needs to understand all the physical properties of water. The person simply needs to know which entity the word “water” picks out in common usage.

In $V_e$, persons understand moral terms, such as “honesty”, in a similar manner. That is, moral terms are understood by persons through direct references instantiated in exemplars. Persons identify moral exemplars through the emotion of admiration. The emotions of admiration and contempt play a foundational role in this theory. $V_e$ posits a process very similar to scientific or empirical investigation. Exemplars are first identified and their traits are studied; then they are continuously further studied to better understand their traits, qualities, etc. The status of an individual as an exemplar can change over time. Below is an informal version that we seek to formalize:

<table>
<thead>
<tr>
<th>Informal Version $V_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$ Agent or person $a$ perceives a person $b$ perform an action $\alpha$. This observation causes the emotion of admiration in $a$.</td>
</tr>
<tr>
<td>$I_2$ $a$ then studies $b$ and seeks to learn what traits (habits/dispositions) $b$ has.</td>
</tr>
</tbody>
</table>

**The Goal**

From the above presentation of $V_e$, we can glean the following distilled requirements that should be present in any formalization.

<table>
<thead>
<tr>
<th>$V_e$ Formalization Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R1) A formalization of emotions, particularly admiration.</td>
</tr>
<tr>
<td>(R2) A representation of traits.</td>
</tr>
<tr>
<td>(R3) A process of learning traits (and not just simple individual actions) from a small number of observations.</td>
</tr>
</tbody>
</table>

**Building the Formalization**

For fleshing out the above requirements and formalizing $V_e$, we will use the deontic cognitive effect calculus ($DCEC$), a computational formal logic. This logic was used previously in (Govindarajulu and Bringsjord 2017a; Govindarajulu et al. 2017) to automate versions of the Doctrine of Double Effect ($DDE$), an ethical principle with deontological and consequentialist components. $DCEC$ has also been used to formalize akrasia (the process of succumbing to temptation to violate moral principles) (Bringsjord et al. 2014). Fragments of $DCEC$ have been used to model highly intensional reasoning processes, such as the false-belief task (Arkoudas and Bringsjord 2008).5

---

4 In her book, Vallor gives her own more detailed and technologically relevant list of seven core elements that can be viewed as common to all members of $V$ (or two what she refers to as “traditions” within virtue ethics). We do not have the space to discuss this list, and show that it fits nicely with our technical work’s emphasis on exemplars and learning therefrom.

5 $DCEC$ is both intensional and intentional. There is a difference between intensional and intensional systems. Broadly speaking, extensional systems are formal systems in which the references and meanings of terms are independent of any context. Intensional systems are formal systems in which meanings of terms are dependent on context, such as the cognitive states of agents, time, etc. Modal logics used for modeling beliefs, desires, and intentions are considered intensional systems. Please see the appendix in (Govindarajulu and Bringsjord 2017a) for a more detailed discussion.
Overview of DCEC

DCEC is a quantified multi-operator\(^6\) modal logic (also known as sorted first-order multi-operator modal logic) that includes the event calculus, a first-order calculus used for commonsense reasoning over time and change (Mueller 2014). This calculus has a well-defined syntax and proof calculus; see Appendix A of (Govindarajulu and Bringsjord 2017a). The proof calculus is based on natural deduction (Gentzen 1935), and includes all the introduction and elimination rules for first-order logic, as well as inference schemata for the modal operators and related structures. As a sorted calculus, DCEC can be regarded analogous to a typed programming language. We show below some of the important sorts used in DCEC. Among these, the Agent, Action, and ActionType sorts are not native to the event calculus.

<table>
<thead>
<tr>
<th>Sort</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Human and non-human actors.</td>
</tr>
<tr>
<td>Time</td>
<td>The Time type stands for time in the domain;</td>
</tr>
<tr>
<td>Event</td>
<td>Used for events in the domain.</td>
</tr>
<tr>
<td>ActionType</td>
<td>Abstract actions instantiated at particular times by actors.</td>
</tr>
<tr>
<td>Action</td>
<td>Events that occur as actions by agents.</td>
</tr>
<tr>
<td>Fluent</td>
<td>Representing states of the world.</td>
</tr>
</tbody>
</table>

Note: actions are events that are carried out by an agent. For any action type \(\alpha\) and agent \(a\), the event corresponding to \(a\) carrying out \(\alpha\) is given by \(\text{action}(a, \alpha)\). For instance, if \(\alpha\) is “running” and \(a\) is “Jack”, \(\text{action}(a, \alpha)\) denotes “Jack is running”.

Syntax The syntax has two components: a first-order core and a modal system that builds upon this core. The figures below show the formal language and inference schemata of DCEC. Commonly used function and relation symbols of the event calculus are included. Any formally defined calculi (e.g. the venerable situation calculus) for modeling commonsense and physical reasoning can be easily switched out in-place of the event calculus.

The modal operators present in the calculus include the standard operators for knowledge \(K\), belief \(B\), desire \(D\), intention \(I\), obligation \(O\) etc. For example, consider \(B(a, t, \phi)\), which says that agent \(a\) believes at time \(t\) the proposition \(\phi\). Here \(\phi\) can in turn be any arbitrary formula.

\(^6\)The full catalogue of available operators exceeds those for belief, desire, and intention, and \(a\) fortiori exceeds the available operators in any standard modal logic designed to formalize e.g. only alethic, epistemic, or deontic phenomena.

Syntax (fragment)

\[ S ::= \text{Agent} \mid \text{ActionType} \mid \text{Action} \mid \text{Event} \mid \text{Fluent} \]

\[ f ::= \text{Agent} \mid \text{ActionType} \rightarrow \text{Action} \]

Inference Schemata The figure below shows a fragment of inference schemata for DCEC. \(I_B\) is an inference schema that let us model idealized agents that have their knowledge and belief closed under the DCEC proof theory. While normal humans are not deductively closed, this lets us model more closely how deliberative agents such as organizations and more strategic actors reason. (Some dialects of cognitive calculi restrict the number of iterations on intensional operators.) \(I_{12}\) states that if an agent \(s\) communicates a proposition \(\phi\) to \(h\), then \(h\) believes that \(s\) believes \(\phi\). \(I_{14}\) dictates how obligations propagate to intentions.

Inference Schemata (fragment)

\[ B(a, t_1, \Gamma), \Gamma \vdash \phi, t_1 < t_2 \quad \frac{[I_B]}{B(a, t_2, \phi)} \]

\[ B(a, t, \phi) \quad \frac{[I_{12}]}{B(h, t, B(s, t_2, \phi))} \]

\[ B(a, t, \phi) \quad \frac{[I_{14}]}{O(a, t, \phi, \chi)} \]

We also define the following inference-schemata-based relationships between expressions in our calculus.

Generalization of Formulae The generalization of a set of formulæ \(\Psi\), is a set of formulæ \(\Phi\) from which any element of \(\Psi\) can be inferred: \(\Phi \vdash \bigwedge \Psi\). This is denoted by \(g(\Psi) = \Phi\).

Generalization of Terms: A term \(x\) is a generalization of a term \(y\) if given any first-order predicate \(P\), from \(P(x)\) we can derive \(P(y)\): \(\{P(x)\} \vdash P(y)\). This is denoted by \(g(y) = x\).

Semantics

DCEC uses proof-theoretic semantics (Gentzen 1935; Francez and Dyckhoff 2010), an approach commonly associated with natural deduction inference systems. Briefly, in this approach, meanings of modal operators are defined via functions over proofs. Specifying semantics then reduces to specifying inference schemata.

Events, Fluents, and Utilities

In the event calculus, fluents represent states of the world. Our formalization of admiration requires a notion of utility for states of the world. Therefore, we assign utilities to fluents through a utility function: \(\mu : \text{Fluent} \times \text{Time} \rightarrow \mathbb{R}\). An event can initiate one or more fluents. Therefore, events can also have a utility associated with them. For an event \(e\) at time \(t\), let \(e^t\) be the set of fluents initiated by the event,
and let \( \mathcal{F}_t \) be the set of fluents terminated by the event. If we are looking up till horizon \( H \), then \( \nu(e, t) \), the total utility of event \( e \) at time \( t \), is:

\[
\nu(e, t) = \sum_{y=t+1}^{H} \left( \sum_{f \in \mathcal{F}_y} \mu(f, y) - \sum_{f \in \mathcal{F}_y} \mu(f, y) \right)
\]

With the calculus given above, we now move on to specifying parts of the formalization \( \mathcal{V}_T \), that is, \( R_1 \), \( R_2 \), and \( R_3 \).

### Defining Admiration

We start with \( R_1 \) and formalize admiration in DCEC. To achieve this, we build upon the OCC model. There are many models of emotion from psychology and cognitive science. Among these, the OCC model (Ortony, Collins, and Clore 1988) has found wide adoption among computer scientists. Note that the model presented by (Ortony, Collins, and Clore 1988) is informal in nature and one formalization of the model has been presented in (Adam, Herzig, and Longin 2009). This formalization is based on propositional modal logic, and while comprehensive and elaborate, is not expressive enough for our modelling, which requires at the least quantification over objects.

In OCC, emotions are short-lived entities that arise in response to events. Different emotions arise based on: (i) whether the consequences to events are positive (desirable) or negative (undesirable); (ii) whether the event has occurred; and (iii) whether the event has consequences for the agent or for another agent. OCC assumes an undefined primitive notion of an agent being pleased or displeased in response to an event. We represent this notion by a predicate \( \Theta \) in our formalization. In OCC, admiration is defined as “(approving of) someone else’s praiseworthy action.” We translate this definition into DCEC as follows. An agent \( a \) is said to admire another agent \( b \)’s action \( \alpha \), if agent \( a \) believes the action is a good action. An action \( \text{action}(b, \alpha) \) is a considered a good action if \( \nu(\text{action}(b, \alpha), t) > 0 \). In OCC, agents can admire only other agents and not themselves. This is captured by the inequality \( a \sim b \). (R1) Admiration in DCEC

\[
\text{holds(admires}(a, b, \alpha), t) \Leftrightarrow \\
\Theta(a, t') \land \\
B \left( \begin{array}{l}
(a \neq b) \land (t' < t) \\
\land \text{happens}(\text{action}(b, \alpha), t') \land \\
\nu(\text{action}(b, \alpha), t) > 0
\end{array} \right)
\]

### Defining Traits

To satisfy \( R_2 \), we need to define traits. We define a situation \( \sigma(t) \) as simply a collection of formulae that describe what fluents hold at a time \( t \), along with other event-calculus constraints and descriptions (sometimes we use \( \sigma(t) \) to represent the conjunction of all the formulae in \( \sigma(t) \)).

\( \nu(\text{action}(a, \alpha), t) \)

(R2) Trait

An agent \( a \) has a situation \( \sigma \) and action type \( \alpha \) as an \( m \)-trait \( (\sigma, \alpha) \) if there are at least \( m \) situations \( \{\sigma_1, \sigma_2, \ldots, \sigma_m\} \) in which instantiations of \( \alpha \) are performed, and \( \sigma \) is the generalization of the situations.

A trait \( (\sigma, \alpha) \) can be represented as single formula:

\[
\tau \equiv \sigma \land \text{happens}(\text{action}(\alpha, a), t)
\]

We introduce a new modal operator \( \text{Trait} \) that can then be applied to the collection of formulae \( \tau \) denoting a trait. \( \text{Trait}(\tau, a) \) says that agent \( a \) has trait \( \tau \). The following inference schema then applies to \( \text{Trait} \):

(R2) Inference Schema for Trait

\[
\text{Trait}(\tau, a) \equiv \{ \sigma_i, \text{happens}(\text{action}(\alpha_i, a), t_i) \}_{i=1}^{n} \}
\]

### Defining Learning of Traits

To address \( R_3 \) we need a definition of what it means for an agent to learn a trait. We start with a learning agent \( l \). An agent \( e \) is identified as an exemplar by \( l \) iff the emotion of admiration is triggered \( n \) times or more by \( e \) in \( l \). This is written down in DCEC as follows (note that admiration can be triggered by different actions):

**Exemplar Definition**

\[
\text{Exemplar}(e, l) \iff \exists t. \exists \alpha. \text{holds(admires}(e, l, \alpha), t)
\]

Once \( e \) is identified as an exemplar, the learner then identifies one or more traits of \( e \) by observing \( e \) over an extended period of time. Let \( l \) believe that \( e \) has a trait \( \tau \); then \( l \) incorporates \( \tau \) as its own trait:

(R3) Learning a Trait

\[
\text{LearnTrait}(l, \tau, t) \Leftrightarrow \exists e \left[ \text{Exemplar}(e, l) \land \\
\text{B}(l, t, \text{Trait}(\tau, e)) \right]
\]

**Example** For instance, if the action type “being truthful” is triggered in situations: “talking with alice”, “talking with bob”, “talking with charlie”; then the trait learned is that “talking with an agent” situation should trigger the “being truthful” action type.

### A Note on Learning Methods

When we look at humans learning virtues by observing others or by reading from texts or other sources, it is not entirely clear how models of learning that have been successful in perception and language processing (e.g. the recent successes of deep learning and statistical learning) can be applied. Learning in V-relevant situations is from one or few instances or in some cases through instruction, and such
learning may not be readily amenable to models of learning which require a large number of examples.

The abstract learning method that we will use is **generalization**, defined previously. See one simple example immediately below:

**Example 1**

\[
\Gamma_1 = \{ \text{talkingWith(jack)} \rightarrow \text{Honesty} \}
\]

\[
\Gamma_2 = \{ \text{talkingWith(jill)} \rightarrow \text{Honesty} \}
\]

**generalization** \( \Gamma = \{ \forall x. \text{talkingWith}(x) \rightarrow \text{Honesty} \} \)

One particularly efficient and well-studied mechanism to realize generalization is **anti-unification**, which has been applied successfully in learning programs from few examples.\(^7\) In anti-unification, we are given a set of expressions \( \{ f_1, \ldots, f_n \} \) and need to compute an expression \( g \) that when substituted with an appropriate term \( \theta \) gives us \( f_i \). For example, if we are given \( \text{hungry}(\text{jack}) \) and \( \text{hungry}(\text{jill}) \), the anti-unification of those terms would be \( \text{hungry}(x) \).

In higher-order anti-unification, we can substitute function symbols and predicate symbols. Here \( P \) is a higher-order variable.

<table>
<thead>
<tr>
<th>Example2</th>
<th>Example3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{likes}(\text{jill}, \text{jack}) )</td>
<td>( \text{likes}(\text{jill}, \text{jack}) )</td>
</tr>
<tr>
<td>( \text{likes}(\text{jill}, \text{im}) )</td>
<td>( \text{loves}(\text{jill}, \text{im}) )</td>
</tr>
<tr>
<td>( \text{likes}(\text{jill}, x) )</td>
<td>( P(\text{jill}, x) )</td>
</tr>
</tbody>
</table>

**Defining Virtuous Person and Virtues**

With the formal machinery in place we finally present formalizations that answer \( Q_1 \) and \( Q_2 \) posed at the outset. An \( n \)-virtuous person or agent \( s \) is an agent that is considered as an exemplar by \( n \) agents:

\( Q_1 \): **Virtuous Person**

\[ V^n \_s(a) \leftrightarrow \exists \alpha^n \_a \cdot \text{Exemplar}(s, \alpha) \]

An \( n \)-virtue is a trait possessed by at least \( n \) virtuous agents:

\( Q_2 \): **Virtue**

\[ G^n \_a(\tau) \leftrightarrow \exists \alpha^n \_a \cdot \text{Trait}(\tau, \alpha) \]

**Implementation & Simulation**

We have extended **ShadowProver**, a quantified modal logic prover for \( DCEC \) used in (Govindaraju and Bringsjord 2017a) to handle the new inference schema and definitional axioms given above. We now show a small simulation in which an agent learns a trait and uses that trait to perform an action. Assume that we have a marketplace where things that are either old or new can be bought and sold. A seller

---

\(^7\)This discipline, known as **inductive programming**, seeks to build precise computer programs from examples (Nienhuys-Cheng and De Wolf 1997). See (Muggleton et al. 2018) for an application in generating human comprehensible programs.

---

\(^8\)See: [https://github.com/naveensundarg/prover/releases/tag/virtue-ethics-simulation](https://github.com/naveensundarg/prover/releases/tag/virtue-ethics-simulation)

**Conclusion & Future Work**

We have presented an initial formalization \( V^2 \_d \) of a virtue ethics theory \( V \_d \) in a calculus that has been used in automating other ethical principles in deontological and consequentialist ethics. Many important questions have to be addressed in future research. Among them are questions about the nature and source of the utility functions that are used in the definitions of emotions. Lacking in our above model is an account of uncertainties and how they interact with virtues.

We plan to leverage an account of uncertainty for a fragment of \( DCEC \) presented in (Govindaraju and Bringsjord 2017b). In future work, we will compare learning traits with work on learning norms (Sarathy, Scheutz, and Malle 2017). The notion of learning we have presented here is quite abstract. In order to handle more complex traits, more sophisticated learning frameworks may have to be considered. Finally, we need to apply this model to more realistic examples and case studies, and implement our theories in realistic robotics architectures (Sarathy et al. 2016).