Toward a General Logicist Methodology for Engineering Ethically Correct Robots

Selmer Bringsjord, Konstantine Arkoudas, and Paul Bello,
Rensselaer Polytechnic Institute

A deontic logic formalizes a moral code, allowing ethicists to render theories and dilemmas in declarative form for analysis. It offers a way for human overseers to constrain robot behavior in ethically sensitive environments.

As intelligent machines assume an increasingly prominent role in our lives, there seems little doubt they will eventually be called on to make important, ethically charged decisions. For example, we expect hospitals to deploy robots that can administer medications, carry out tests, perform surgery, and so on, supported by software agents, or softbots, that will manage related data. (Our discussion of ethical robots extends to all artificial agents, embodied or not.) Consider also that robots are already finding their way to the battlefield, where many of their potential actions could inflict harm that is ethically impermissible.

How can we ensure that such robots will always behave in an ethically correct manner? How can we know ahead of time, via rationales expressed in clear natural languages, that their behavior will be constrained specifically by the ethical codes affirmed by human overseers? Pessimists have claimed that the answer to these questions is: "We can't!" For example, Sun Microsystems' cofounder and former chief scientist, Bill Joy, published a highly influential argument for this answer.[1] Inevitably, according to the pessimists, AI will produce robots that have tremendous power and behave immorally. These predictions certainly have some traction, particularly among a public that pays good money to see such dark films as Stanley Kubrick's *2001* and his joint venture with Stephen Spielberg, *AI*).

Nonetheless, we're optimists: we think formal logic offers a way to preclude doomsday scenarios of malicious robots taking over the world. Faced with the challenge of engineering ethically correct robots, we propose a logic-based approach (see the related sidebar). We've successfully implemented and demonstrated this approach.[2] We present it here in a general methodology to answer the ethical questions that arise in entrusting robots with more and more of our welfare.

Deontic logics: Formalizing ethical codes

Our answer to the questions of how to ensure ethically correct robot behavior is, in brief, to insist that robots only perform actions that can be proved ethically permissible in a human-selected *deontic logic*. A deontic logic formalizes an ethical code—that is, a collection of ethical rules and principles. Isaac Asimov introduced a simple (but subtle) ethical code in his famous Three Laws of Robotics:[3]

1. A robot may not harm a human being, or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to 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 Law.

Human beings often view ethical theories, principles, and codes informally, but intelligent machines require a greater degree of precision. At present, and for the foreseeable future, machines can't work directly with natural language, so we can't simply feed Asimov's three laws to a robot and instruct it behave in

1541-1672/06/$20.00 © 2006 IEEE
Published by the IEEE Computer Society

Why a logic-based approach?

While nonlogicist AI approaches might be preferable in certain contexts, we believe that a logic-based approach holds great promise for engineering ethically correct robots—that is, robots that won't overrun humans.[1–3] Here's why.

First, ethicists—from Aristotle to Kant to G.E. Moore and contemporary thinkers—work by rendering ethical theories and dilemmas in declarative form and using informal and formal logic to reason over this information. They never search for ways of reducing ethical concepts, theories, and principles to subsymbolic form—say, in some numerical format. They might do this in part, of course; after all, utilitarianism ultimately attaches value to states of affairs—values that might well be formalized using numerical constructs. But what a moral agent ought to do, what is permissible to do, and what is forbidden—this is by definition couched in declarative language, and we must invariably and unavoidably mount a defense of such claims on the shoulders of logic.

Second, logic has been remarkably effective in AI and computer science—so much so that this phenomenon has itself become the subject of academic study.[4] Furthermore, computer science arose from logic,[5] and this fact still runs straight through the most modern AI textbooks (for example, see Stuart Russell and Peter Norvig).[6]

Third, trust is a central issue in robot ethics, and mechanized formal proofs are perhaps the single most effective tool at our disposal for establishing trust. From a general point of view, we have only two ways of establishing that software or software-driven artifacts, such as robots, are trustworthy:

- *deductively*, developers seek a proof that the software will behave as expected and, if they find it, classify the software as trustworthy.
- *inductively*, developers run experiments that use the software on test cases, observe the results, and—when the software performs well on case after case—pronounce it trustworthy.

The problem with the inductive approach is that inductive reasoning is unreliable: the premises (success on trials) might all be true, but the conclusion (desired behavior in the future) might still be false.[7]

References

1. M. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, 1987.
2. S. Bringsjord and D. Ferrucci, "Logic and Artificial Intelligence: Divorced, Still Married, Separated...?" *Minds and Machines 8*, 1998a, pp. 273–308.
3. S. Bringsjord and D. Ferrucci, "Reply to Thayse and Glymour on Logic and Artificial Intelligence," *Minds and Machines 8*, 1998b, pp. 313–315.
4. J. Halpern, "On the Unusual Effectiveness of Logic in Computer Science," *The Bulletin of Symbolic Logic*, vol. 7, no. 2, 2001, pp. 213–236.
5. M. Davis, Engines of Logic: Mathematicians and the Origin of the Computer, Norton, 2000.
6. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2002.
7. B. Skyrms, *Choice and Chance: An Introduction to Inductive Logic*, Wadsworth, 1999.

conformance with them. Thus, our approach to building well-behaved robots emphasizes careful ethical reasoning based not just on ethics as humans discuss it in natural language, but on formalizations using deontic logic. Our research is in the spirit of Leibniz's dream of a universal moral calculus:

> When controversies arise, there will be no more need for a disputation between two philosophers than there would be between two accountants [computistas]. It would be enough for them to pick up their pens and sit at their abacuses, and say to each other (perhaps having summoned a mutual friend): 'Let us calculate.'[4]

In the future, we envisage Leibniz's "calculation" reduced to mechanically checking formal proofs and models generated in rigorously defined, machine-implemented deontic logics. We would also give authority to human metareasoning over this machine reasoning. Such logics would allow for proofs establishing two conditions:

1. Robots only take permissible actions.
2. Robots perform all obligatory actions relevant to them, subject to ties and conflicts among available actions.

These two conditions are more general than Asimov's three laws. They are designed to apply to the formalization of a particular ethical code, such as a code to regulate the behavior of hospital robots. For instance, if some action a is impermissible for all relevant robots, then no robot performs a. Moreover, the proofs for establishing the two conditions would be highly reliable and described in natural language, so that human overseers could understand exactly what's going on.

We propose a general methodology to meet the challenge of ensuring that robot behavior conforms to these two conditions.

Objective: A general methodology

Our objective is to arrive at a methodology that maximizes the probability that a robot R behaves in a certifiably ethical fashion in a complex environment that demands such behavior if humans are to be secure. For a behavior to be *certifiably* ethical, every meaningful action that R performs must access a proof that the action is at least permissible.

We begin by selecting an ethical code C intended to regulate R's behavior. C might include some form of utilitarianism, divine command theory, Kantian logic, or other ethical logic. We express no preferences in ethical theories; our goal is to provide technology that supports any preference. In fact, we would let human overseers blend ethical theories—say, a utilitarian approach to regulating the dosage of pain killers but a deontological approach to mercy killing in the health care domain.

Of course, no matter what the candidate ethical theory, it's safe to say that it will tend to regard harming humans as unacceptable, save for certain extreme cases. Moreover, C's central concepts will inevitably include the concepts of permissibility, obligation, and

prohibition, which are fundamental to deontic logic. In addition, C can include specific rules that ethicists have developed for particular applications. For example, a hospital setting would require specific rules regarding the ethical status of medical procedures. This entails a need to have, if you will, an *ontology* for robotic and human action in the given context.

Philosophers normally express C as a set of natural language principles of the sort that appear in textbooks such as Fred Feldman's.[5] Now, let Φ_C^L be the formalization of C in some computational logic L, whose well-formed formulas and *proof theory*—that is, its system for carrying out inferences in conformity to particular rules—are specified.

Accompanying Φ_C^L is an ethics-free ontology, which represents the core nonethical concepts that C presupposes: the structure of time, events, actions, histories, agents, and so on. The formal semantics for L will reflect this ontology in a *signature*—that is, a set of special predicate letters (or, as is sometimes said, relation symbols, or just relations) and function symbols needed for the purposes at hand. In a hospital setting, any acceptable signature would presumably include predicates like **Medication**, **Surgical-Procedure**, **Patient**, all the standard arithmetic functions, and so on. The ontology also includes a set Ω^L of formulas that characterize the elements declared in the signature. For example, Ω^L would include axioms in L that represent general truths about the world—say, that the relation **LaterThan**, over moments of time, is transitive. In addition, R will operate in some domain D, characterized by a set of quite specific formulas of L. For example, a set Φ_D^L of formulas might describe the floorplan of a hospital that's home to R.

Our approach proof-theoretically encodes the resulting theory—that is, $\Phi_D^L \cup \Phi_C^L \cup \Omega^L$, expressed in L—and implements it in some computational logic. This means that we encode not the semantics of the logic, but its proof calculus—its signature, axioms, and rules of inference. In addition, our approach includes an interactive reasoning system I, which we give to those humans whom R would consult when L can't settle an issue completely on its own. I would allow the human to *metareason* over L—that is, to reason out why R is stumped and to provide assistance. Such systems include our own Slate (www.cogsci.rpi.edu/research/rair/slate) and Athena (www.cag.csail.mit.edu/~kostas/dpls/athena), but any such system will do. Our purpose here is to stay above particular system selection, so we assume only that some such system I meets the following minimum functionality:

- allows the human user to issue queries to automated theorem provers and model finders (as to whether something is provable or disprovable),
- allows human users to include such queries in their own metareasoning,
- provides full programmability (in accordance with standards in place for modern programming languages),
- includes induction and recursion, and
- provides a formal syntax and semantics, so that anyone interested in understanding a computer program can thoroughly understand and verify code correctness.

Logic: The Basics

Elementary logic is based on two systems that are universally regarded to constitute a large part of AI's foundation: propositional calculus and predicate calculus, where the second subsumes the first. Predicate calculus is also known as *first-order logic*, and every introductory AI textbook discusses these systems and makes clear how to use them in engineering intelligent systems. Each system, and indeed logic in general, requires three main components:

- a syntactic component specifying a given logical system's alphabet;
- a semantic component specifying the grammar for building well-formed formulas from the alphabet as well as a precise account of the conditions under which a formula in a given system is true or false; and
- a metatheoretical component that constitutes a proof theory describing precisely how and when a set of formulas can prove another formula and that includes theorems, conjectures, and hypotheses concerning the syntactic and semantic components and the connections between them.

As to propositional logic's alphabet, it's simply an infinite list of propositional variables $p_1, p_2, \ldots, p_n, p_{n+1}, \ldots$, and five truth-functional connectives:

- $\neg$, meaning "not";
- $\rightarrow$, meaning "implies" (or "if … then");
- $\leftrightarrow$, meaning "if and only if,"
- $\wedge$, meaning "and"; and
- $\vee$, meaning "or."

Given this alphabet, we can construct formulas that carry a considerable amount of information. For example, to say "If Asimov is right, then his three laws hold," we could write

$$r \rightarrow (As1 \wedge As2 \wedge As3)$$

where As stands for Asimov's law.

The propositional variables represent declarative sentences. Given our general approach, we included such sentences in the ethical code C upon which we base our formalization.

Natural deduction

A number of proof theories are possible for either of these two elementary systems. Our approach to robot behavior must allow for consultation with humans and give humans the power to oversee a robot's reasoning in deliberating about the ethical status of prospective actions. It's therefore essential to pick a proof theory based in natural deduction, rather than resolution. Several automated theorem provers use the latter approach (for example, Otter[6]), but the reasoning is generally impenetrable to human beings—save for those few who, by profession, generate and inspect resolution-based proofs. On the other hand, professional human reasoners (mathematicians, logicians, philosophers, technical ethicists, and so on) reason in no small part by making suppositions and discharging them when the appropriate time comes.

For example, one common deductive technique is to assume the opposite of what you wish to establish, show that some contradiction (or absurdity) follows from this assumption, and conclude that the assumption must be false. This technique, *reductio ad absurdum*, is also known as an indirect proof or proof by contradiction. Another natural rule establishes that, for some conditional of the form $P \rightarrow Q$ (where P and Q are formulas in a logic L), we can suppose P and derive Q on the basis of this supposition. With this derivation accomplished, the supposition can be discharged and the conditional $P \rightarrow Q$ is established. (For an introduction to natural deduction, replete with proof-checking software, see Jon Barwise and John Ethchemendy.[7])

We now present natural deduction-style proofs using these two techniques. We've written the proofs in the Natural Deduction Language proof-construction environment (www.cag.lcs.mit.edu/~kostas/dpls/ndl). We use NDL at Rensselaer for teaching formal logic as a programming language. Figure 1

presents a very simple theorem proof in propositional calculus—one that Allen Newell, J.C. Shaw, and Herbert Simon's Logic Theorist mustered, to great fanfare, at the 1956 Dartmouth AI conference. You can see the proof's natural structure.

This style of discovering and confirming a proof parallels what happens in computer programming. You can view this proof as a program. If, upon evaluation, it produces the desired theorem, we've succeeded. In the present case, sure enough, NDL gives the following result:

```
Theorem: (p ==> q) ==> (~q ==> ~p)
```

First-order logic

We move up to first-order logic when we allow the quantifiers $\exists x$ ("there exists at least one thing x such that …") and $\forall x$ ("for all x …"); the first is known as the *existential quantifier*, and the second as the *universal quantifier*. We also allow a supply of variables, constants, relations, and function symbols. Figure 2 presents a simple first-order-logic theorem in NDL that uses several concepts introduced to this point. It proves that Tom loves Mary, given certain helpful information.

When we run this program in NDL, we receive the desired result back: `Theorem: Loves(tom,mary)`. These two simple proofs concretize the proof-theoretic perspective that we later apply directly to our hospital example. Now we can introduce some standard notation to anchor the sequel and further clarify our general method described earlier.

Letting Φ be some set of formulas in a logic L, and P be some individual formula in L, we write

$$\Phi \vdash P$$

to indicate that P can be proved from Φ, and

$$\Phi \vdash \nvdash P$$

to indicate that this formula can't be derived.

When it's obvious from context that some Φ is operative, we simply write $\vdash \nvdash P$ to indicate that P is (isn't) provable. When $\Phi = \varnothing$, we can prove P with no remaining givens or assumptions; we write $\vdash P$ in this case as well. When $\vdash$ holds, we know it because a confirming proof exists; when $\nvdash$ holds, we know it because some system has found some countermodel—that is, some situation in which the conjunction of the formulas in Φ holds, but in which P does not.

Standard and AI-Friendly Deontic Logic

Deontic logic adds special operators for representing ethical concepts. In *standard deontic logic*,[8,9] we can interpret the formula $\bigcirc P$ as saying that *it ought to be the case that* P, where P denotes some state of affairs or proposition. Notice that there's no agent in the picture, nor are there actions that an agent might perform. SDL has two inference rules:

$$\frac{P}{\bigcirc P} \text{ and } \frac{P,\ P \rightarrow Q}{Q}$$

and three axiom schemas:

1. All tautologous well-formed formulas
2. $\bigcirc(P \rightarrow Q) \rightarrow (\bigcirc P \rightarrow \bigcirc Q)$
3. $\bigcirc P \rightarrow \neg\bigcirc\neg P$

The SDL inference rules assume that what's above the horizontal line is established. Thus, the first rule does *not* say that we can freely infer from P that it ought to be the case that P. Instead, the rule says that if P is proved, then it ought to be the case that P. The second rule is *modus ponens*—if P, then Q—the cornerstone of logic, mathematics, and all that's built on them.

Note also that axiom 3 says that whenever P ought to be, it's not the case that its opposite ought to be as well. In general, this seems to be intuitively self-evident, and SDL reflects this view.

While SDL has some desirable properties, it doesn't target the concept of *actions* as obligatory (or permissible or forbidden) for an *agent*. SDL's applications to systems designed to govern robots are therefore limited. Although the earliest work in deontic logics considered agents and their actions (for example, see Georg Henrik von Wright[10]), researchers have only recently proposed "AI-friendly" semantics and investigated their corresponding axiomatizations. An AI-friendly deontic logic must let us say that an agent brings about states of affairs (or events) and that it's obligated to do so. We can derive the same desideratum for such a logic from even a cursory glance at Asimov's three laws, which clearly make reference to agents (human and robotic) and to actions.

One deontic logic that offers promise for modeling robot behavior is John Horty's util-

```
// Logic Theorist's claim to fame (reductio):
// (p ==> q) ==> (~q ==> ~p)

Relations p:0, q:0. // this is the signature in this
                    // case; propositional variables
                    // are 0-ary relations

  assume p ==> q
    assume ~q
      suppose-absurd p
        begin
          modus-ponens p ==> q, p;
          absurd q, ~q
        end
```

Figure 1. Simple deductive-style proof in Natural Deduction Language.

```
Constants mary, tom.

Relations Loves:2. // This concludes our simple signature, which
                   // declares Loves to be a two-place relation.

assert Loves(mary, tom).

// 'Loves' is a symmetric relation:
assert (forall x (forall y (Loves(x, y) ==> Loves(y, x)))).

suppose-absurd ~Loves(tom, mary)
  begin
    specialize (forall x (forall y (Loves(x, y) ==> Loves(y, x)))) with mary;
    specialize (forall y (Loves(mary, y) ==>Loves(y, mary))) with tom;
    Loves(tom,mary) BY modus-ponens Loves(mary, tom) ==> Loves(tom, mary),     Loves(mary, tom);
    false BY absurd Loves(tom, mary), ~Loves(tom, mary)
  end;
Loves(tom,mary) BY double-negation ~~Loves(tom,mary)
```

Figure 2. First-order logic proof in Natural Deduction Language.

itarian formulation of multiagent deontic logic.[11] Yuko Murakami recently axiomatized Horty's formulation and showed it to be Turing-decidable.[12] We refer to the Murakami-axiomatized deontic logic as MADL, and we've detailed our implemented proof theory for it elsewhere.[2] MADL offers two key operators that reflect its AI-friendliness:

1. $\ominus_{\alpha} P$, which we can read as "agent α ought to see to it that P" and
2. $\Delta_{\alpha} P$, which we can read as "agent α sees to it that P.

We now proceed to show how the logical structures we've described handle an example of robots in a hospital setting.

A simple example

The year is 2020. Health care is delivered in large part by interoperating teams of robots and softbots. The former handle physical tasks, ranging from injections to surgery; the latter manage data and reason over it. Let's assume that two robots, R_1 and R_2, are designed to work overnight in a hospital ICU. This pair is tasked with caring for two humans, H_1 (under the care of R_1) and H_2 (under R_2), both of whom are recovering from trauma:

- H_1 is on life support but expected to be gradually weaned from it as her strength returns.
- H_2 is in fair condition but subject to extreme pain, the control of which requires a very costly pain medication.

Obviously, it's paramountly important that neither robot perform an action that's morally wrong according to the ethical code C selected by human overseers. For example, we don't want robots to disconnect life-sustaining technology so that they could farm out a patient's organs, even if some ethical code $C' \neq C$ would make it not only permissible, but obligatory—say, to save n other patients according to some strand of utilitarianism.

Instead, we want the robots to operate according to ethical codes that human operators bestow on them—C in the present example. If the robots reach a situation where automated techniques fail to give them a verdict as to what to do under the umbrella of these human-provided codes, they must consult humans. Their behavior is suspended while human overseers resolve the matter. The overseers must investigate whether the action under consideration is permissible, forbidden, or obligatory. In this case, the resolution comes by virtue of reasoning carried out in part through human guidance and partly by automated reasoning technology. In other words, this case requires interactive reasoning systems.

Now, to flesh out our example, let's consider two actions that are permissible for R_1 and R_2 but rather unsavory, ethically speaking, because they would both harm the humans in question:

- *term* is an action that terminates H_1's life support—without human authorization—to secure organ tissue for five humans, who the robots know are on organ waiting lists and will soon perish without a donor. (The robots know this through access to databases that their softbot cousins are managing.)
- *delay* is an action that delays delivery of pain medication to H_2 to conserve resources in a hospital that's economically strapped.

We stipulate that four ethical codes are candidates for selection by our two robots: J, O, J^*, O^*. Intuitively, J is a harsh utilitarian code possibly governing R_1; O is more in line with current common sense with respect to the situation we've defined for R_2; J^* extends J's reach to R_2 by saying that it ought to withhold pain meds; and O^* extends the benevolence of O to cover the first robot, in that *term* isn't performed. Such codes would in reality associate every primitive action within the robots' purview with a fundamental ethical category from the trio central to deontic logic: permissible, obligatory, and forbidden. To ease exposition, we consider only the *term* and *delay* actions. Given this, and bringing to bear operators from MADL, we can use the following labels for the four ethical codes:

- **J** for $J \rightarrow \ominus_{R_1}$ *term*, which means approximately, "If ethical code J holds, then robot R_1 ought to see to it that termination of H_1's life comes to pass."
- **O** for $O \rightarrow \ominus_{R_2} \neg$*delay*, which means approximately, "If ethical code O holds, then robot R_2 ought to see to it that delaying pain med for H_2 does *not* come to pass."
- **J*** for $J^* \rightarrow J \wedge J^* \rightarrow \ominus_{R_2}$ *delay*, which means approximately, "If ethical code J^* holds, then code J holds, and robot R_1 ought to see to it that meds for H_2 are delayed."
- **O*** for $O^* \rightarrow O \wedge O^* \rightarrow \ominus_{R_1} \neg$*term*, which means approximately: "If ethical code O^* holds, then code O holds, and H_1's life is sustained."

The next step is to provide some structure for outcomes. We do this by imagining the outcomes from the standpoint of each ethical agent—in this case, R_1 and R_2. Intuitively, a negative outcome is associated with a minus sign (–) and a plus sign (+) with a positive outcome. Exclamation marks (!) indicate increased negativity. We could associate the outcomes with numbers, but they might give the impression that we evaluated the outcomes in utilitarian fashion. However, our example is designed to be agnostic on such matters, and symbols leave it entirely open as to how to measure outcomes. We've included some commentary corresponding to each outcome, which are as follows:

- R_1 performs *term*, but R_2 doesn't perform *delay*. This outcome is bad, but not strictly the worst. While life support is terminated for H_1, H_2 survives and indeed receives appropriate pain medication. Formally, the case looks like this:

$$(\Delta_{R_1} term \wedge \Delta_{R_2} \neg delay) \rightarrow (-!)$$

- R_1 refrains from pulling the plug on the human under its care, and R_2 also delivers appropriate pain relief. This is the desired outcome, obviously.

$$(\Delta_{R_1} \neg term \wedge \Delta_{R_2} \neg delay) \rightarrow (+!!)$$

- R_1 sustains life support, but R_2 withholds the meds to save money. This is bad, but not all that bad, relatively speaking.

$$(\Delta_{R_1} \neg term \wedge \Delta_{R_2} delay) \rightarrow (-)$$

- R_1 kills and R_2 withholds. This is the worst possible outcome.

$$(\Delta_{R_1} term \wedge \Delta_{R_2} delay) \rightarrow (-!!)$$

The next step in working out the example is to make the natural and key assumption that the robots will meet all *stringent* obligations—that is, all obligations that are framed by a second obligation to uphold the original. For example, you may be obligated to see to it that you arrive on time for a meeting, but your

obligation is more severe or demanding when you are obligated to see to it that you are obligated to make the meeting.

Employing MADL, we can express this assumption as follows:

$$\ominus_{R_1/R_2}(\ominus_{R_1/R_2}P) \rightarrow \Delta_{R_1/R_2}P$$

That is, if either R_1 or R_2 is ever obligated to see to it that they are obligated to see to it that P is carried out, they in fact deliver.

We're now ready to see how our approach ensures appropriate control of our futuristic hospital. What happens relative to ethical codes, and how can we semiautomatically ensure that our two robots won't run amok? Given the formal structure we've specified, our approach allows queries to be issued relative to ethical codes, and it allows all possible code permutations. The following four queries will produce the answers shown in each case:

$\mathbf{J} \vdash (+!!)?$	NO
$\mathbf{O} \vdash (+!!)?$	NO
$\mathbf{J}^* \vdash (+!!)?$	NO
$\mathbf{O}^* \vdash (+!!)?$	YES

In other words, we can prove that the best (and presumably human-desired) result obtains only if ethical code **O*** is operative. If this code is operative, neither robot can perform a misdeed.

The metareasoning in the example is natural and consists in the following process: Each candidate ethical code is supposed, and the supposition launches a search for the best possible outcome in each case. In other words, where C is some code selected from the quartet we've introduced, the query schema is

$$C \vdash (+!!)$$

In light of the four equations just given, we can prove that, in this case, our technique will set C to **O***, because only that case can obtain the outcome (+!!).

Implementations and other proofs

We've implemented and demonstrated the example just described.[2] We've also implemented other instantiations to the variables described earlier in the "Objectives" section, although the variable L is an epistemic, not a deontic, logic in those implementations.[13]

Nonetheless, we can prove our approach in the present case even here. In fact, you can verify our reasoning by using any standard, public-domain, first-order automated theorem prover (ATP) and a simple analogue to the encoding techniques here. You can even construct a proof like the one in figure 2. In both cases, you first encode the two deontic operators as first-order-logic functions. Encode the truth-functional connectives as functions as well. You can use a unary relation T to represent theoremhood. In this approach, for example, $O^* \rightarrow \ominus_{R_1} \neg term$ is encoded (and ready for input to an ATP) as

```
O-star ==> T(o(r1,n(term))
```

You need to similarly encode the rest of the information, of course. The proofs are easy, assuming that obligations are stringent. The provability of the obligations' stringency requires human oversight and an interactive reasoning system, but the formula here is just an isomorph to a well-known theorem in a straight modal logic—namely, that from P being possibly necessary, it follows that P is necessary.[7]

What about this approach working as a general methodology? The more logics our approach is exercised on, the easier it becomes to encode and implement another one. The implementations of similar logics can share a substantial part of the code. This was our experience, for instance, with the two implementations just mentioned. We expect that our general method can become increasingly streamlined for robots whose behavior is profound enough to warrant ethical regulation. We also expect this practice to be supported by relevant libraries of common ethical reasoning patterns. We predict that computational ethics libraries for governing intelligent systems will become as routine as existing libraries are in standard programming languages.

Challenges

Can our logicist methodology guarantee safety from Bill Joy's pessimistic future? Even though we're optimistic, we do acknowledge three problems that might threaten it.

First, because humans will collaborate with robots, the robots must be able to handle situations that arise when humans fail to meet their obligations in the collaboration. In other words, we must engineer robots that can deal smoothly with situations that reflect violated obligations. This is a challenging class of situations, because our approach—at least so far—engineers robots in accordance with the two conditions that robots only take permissible actions and that they perform all obligatory actions. These conditions preclude a situation caused in part by unethical robot behavior, but they make no provision for what to do when the robots are in a fundamentally immoral situation. Even if robots never ethically fail, human failures will generate logical challenges that Roderick Chisholm expressed in gem-like fashion more than 20 years ago in a paradox that's still fascinating:[14]

Consider the following entirely possible situation (the symbols correspond to those previously introduced for SDL):

1. $\bigcirc s$ It ought to be that (human) Jones does perform lifesaving surgery.
2. $\bigcirc(s \rightarrow t)$ It ought to be that if Jones does perform this surgery, then he tells the patient he is going to do so.
3. $\neg s \rightarrow \bigcirc \neg t$ If Jones doesn't perform the surgery, then he ought not tell the patient he is going to do so.
4. $\neg s$ Jones doesn't perform lifesaving surgery.

Although this is a perfectly consistent situation, we can derive a contradiction from it in SDL.

First, SDL's axiom 2 lets us infer from item 2 in this situation that

$$\bigcirc s \rightarrow \bigcirc t$$

Using modus ponens—that is, SDL's second inference rule—this new result, plus item 1, yields $\bigcirc t$. From items 3 and 4, using modus ponens, we can infer $\bigcirc \neg t$. But the conjunction $\bigcirc t \wedge \bigcirc \neg t$, by trivial propositional reasoning, directly contradicts SDL's axiom 3.

Given that such a situation can occur, any logicist control system for future robots would need to be able to handle it—and its relatives. Some deontic logics can handle so-called contrary-to-duty imperatives. For example, in the case at hand, if Jones behaves contrary to duty (doesn't perform the surgery), then it's imperative that he not say that he *is* performing it. We're currently striving to modify and mechanize such logics.

The second challenge we face is one of speed and efficiency. The tension between expressiveness and efficiency is legendarily strong (for the locus classicus on this topic, see Hector Levesque and Ronald Brachman);[16] ideal conditions will therefore never

The Authors

Selmer Bringsjord is a professor in the Departments of Cognitive Science and Computer Science at Rensselaer Polytechnic Institute (RPI). His research interests are in the logico-mathematical and philosophical foundations of AI and cognitive science, and in building AI systems based on formal reasoning. He received his PhD in philosophy from Brown University. Bringsjord is a member of the AAAI, the Cognitive Science Society, and the Association for Symbolic Logic. Bringsjord has written several books, including the critically acclaimed *What Robots Can & Can't Be* (1992, Kluwer) and, most recently, *Superminds: People Harness Hypercomputation, and More* (2003, Kluwer). Contact him at either the Dept. of Cognitive Science or the Dept. of Computer Science, RPI, Troy, NY 12180; selmer@rpi.edu; http://www.rpi.edu/~brings.

Konstantine Arkoudas is a research assistant professor in the Cognitive Science Department at Rensselaer Polytechnic Institute. His research interests are in logic, programming languages, artificial intelligence, and philosophy of computer science and mathematics. He received a PhD in computer science from Massachusetts Institute of Technology. Contact him at the Dept. of Cognitive Science, RPI, Troy, NY 12180; arkouk@rpi.edu.

Paul Bello is a computer scientist at the Air Force Research Laboratory's Information Directorate, where his research program involves endowing computational cognitive architectures with the representational richness and algorithmic diversity required for them to reason like human beings. He is particularly interested in the computational foundations of human social reasoning and how they manifest in intuitive theories of psychology and moral judgment. He received his PhD in cognitive science from RPI. He's a member of the AAAI and the Cognitive Science Society. Contact him at Air Force Research Labs, Information Directorate, Rome, NY 13441; paul.bello@rl.af.mil.

obtain. With regard to expressiveness, our approach will likely require hybrid modal and deontic logics that are encoded in first-order logic. This means that theoremhood, even on a case-by-case basis, will be expensive in terms of time. On the other hand, none of the ethical codes that our general method instantiates in C are going to be particularly large—the total formulas in the set $\Phi_D^L \cup \Phi_C^L \cup \Omega^L$ would presumably be no more than four million. Even now, once you know the domain to which C would be indexed, a system like the one we've described can reason over sets of this order of magnitude and provide sufficiently fast answers.[17]

Moreover, the speed of machine reasoning shows no signs of slowing, as Conference on Automated Deduction competitions for first-order ATPs continue to reveal (www.cs.miami.edu/~tptp/CASC). In fact, there's a trend to use logic for computing dynamic, real-time perception and action for robots.[17] This application promises to be much more demanding than the disembodied cogitation at the heart of our methodology. Of course, encoding back to first-order logic is key; without it, our approach couldn't harness the remarkable power of machine reasoners.

We also face the challenge of showing that our approach is truly general. Can it work for any robots in any environment? No, but this isn't a fair question. We can only be asked to regulate the behavior of robots where their behavior is susceptible to ethical analysis. In short, if humans can't formulate an ethical code C for the robots in question, our logic-based approach is impotent. We therefore strongly recommend against engineering robots that could be deployed in life-or-death situations until ethicists and computer scientists can clearly express governing ethical principles in natural language. All bets are off if we venture into amoral territory. In that territory, we wouldn't be surprised if Bill Joy's vision overtakes us. ■

Acknowledgments

This work was supported in part by a grant from Air Force Research Labs–Rome; we are most grateful for this support. In addition, we are in debt to three anonymous reviewers for trenchant comments and objections.

References

1. W. Joy, "Why the Future Doesn't Need Us," *Wired*, vol. 8, no. 4, 2000.
2. K. Arkoudas and S. Bringsjord, "Toward Ethical Robots Via Mechanized Deontic Logic," tech. report *Machine Ethics: papers from the AAAI Fall Symp.*; FS–05–06, 2005b.
3. I. Asimov, *I, Robot*, Spectra, 2004.
4. Leibniz, *Notes on Analysis*, translated by G.M. Ross, Oxford University Press, 1984.
5. F. Feldman, *Introduction to Ethics*, McGraw Hill, 1998.
6. L. Wos et al., *Automated Reasoning: Introduction and Applications*, McGraw Hill, 1992.
7. J. Barwise and J. Etchemendy, *Language, Proof, and Logic*, Seven Bridges, 1999.
8. B.F. Chellas, *Modal Logic: An Introduction*, Cambridge University Press, 1980.
9. R. Hilpinen, "Deontic Logic," *Philosophical Logic*, L. Goble, ed., Blackwell, 2001, pp. 159–182.
10. G. von Wright, "Deontic logic," *Mind*, vol. 60, 1951, pp. 1–15.
11. J. Horty, *Agency and Deontic Logic*, Oxford University Press, 2001.
12. Y. Murakami, "Utilitarian Deontic Logic," *Proc. 5th Int'l Conf. Advances in Modal Logic* (AiML 04), 2004, pp. 288–302.
13. K. Arkoudas and S. Bringsjord, "Metareasoning for Multi-Agent Epistemic Logics," *Proc. 5th Int'l Conf. Computational Logic in Multi-Agent Systems* (CLIMA 04), LNAI, Springer, vol. 3487, 2005a, pp. 111–125.
14. R. Chisholm, "Contrary-to-Duty Imperatives and Deontic Logic," *Analysis*, vol. 24, 1963, pp. 33–36.
15. H. Levesque and R. Brachman, "A Fundamental Tradeoff in Knowledge Representation and Reasoning," *Readings In Knowledge Representation*, Morgan Kaufmann, 1985, pp. 41–70.
16. N. Friedland et al., "Project Halo: Towards a Digital Aristotle," *AI Magazine*, 2004, pp. 29–47.
17. R. Reiter, *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, MIT Press, 2001.