Minds & Machines (2011) 21:261–274
DOI 10.1007/s11023-011-9229-z

Computational Meta-Ethics

Towards the Meta-Ethical Robot

Gert-Jan C. Lokhorst

Received: 4 November 2009 / Accepted: 17 November 2010 / Published online: 2 February 2011

Abstract It has been argued that ethically correct robots should be able to reason about right and wrong. In order to do so, they must have a set of do's and don'ts at their disposal. However, such a list may be inconsistent, incomplete or otherwise unsatisfactory, depending on the reasoning principles that one employs. For this reason, it might be desirable if robots were to some extent able to reason about their own reasoning—in other words, if they had some meta-ethical capacities. In this paper, we sketch how one might go about designing robots that have such capacities. We show that the field of computational meta-ethics can profit from the same tools as have been used in computational metaphysics.

Keywords Automated moral reasoning · Computational meta-ethics

Introduction

A continually growing number of computers/robots is being deployed on the battlefield, in hospitals, in law enforcement, in electronic business negotiation and other ethically sensitive areas. It is desirable that the computers/robots in these areas behave in an ethically correct manner. It might be argued that they can only be guaranteed to do so if they can reason about right and wrong on the basis of a set of moral standards: only "explicit ethical agents" (as they have been called; see Moor 2006) can be expected to behave in an ethically correct manner. However, sets of moral standards can be inconsistent, incomplete, or inappropriate in view of other sets of standards; it would therefore desirable that robots equipped with such

G.-J. C. Lokhorst (✉)
Section of Philosophy, Faculty of Technology, Policy and Management,
Delft University of Technology, P.O. Box 5015, 2600 GA Delft, The Netherlands
e-mail: g.j.c.lokhorst@tudelft.nl

standards were to some extent able to reason about them—in other words, if they had some capacity for meta-ethical reasoning.

These considerations suggest that when one wants to design ethically correct robots one should not only explore the field of *automated moral reasoning*, but also the field of *automated moral meta-reasoning*, in the sense of reasoning about moral reasoning. The development track that we foresee is roughly as follows: informal moral reasoning (reasoning about obligations, permissions and prohibitions)—formal moral reasoning (deontic logic)—automated moral reasoning (computational deontic logic)—automated reasoning about automated moral reasoning (computational meta-ethics).

Along this development track, the area of *automated moral reasoning* has already received some attention. Computational deontic logic has been discussed in the context of computer-supported computer ethics (Hoven and Lokhorst 2002). This work has been used in books on the design of moral machines that can distinguish right from wrong (Wallach and Allen 2008) and engineering autonomous (unsupervised) moral robots that are capable of carrying out lethal behavior (Arkin 2009). Similar research has been described in papers on moral reasoning by ethically correct robots (Arkoudas et al. 2005; Bringsjord et al. 2006).

However, the subject of *automated moral meta-reasoning* has not received much attention so far. We intend to set some cautious first steps in this area. We are interested in machines that reason about moral reasoning—meta-ethical robots—for two reasons. First, it is an intellectual challenge to think about how one might go about constructing such machines, which might be called philosophical in the sense that philosophers (inspired by Carnap 1937) are sometimes fond of saying: “anything you can do, I can do *meta*.” Second, it seems important for practical applications. Ethical robots need ethical standards. However, these standards may need examining, either during the design process or in the deployment stage. To the extent that meta-ethical reasoning can be delegated to computers/robots, the results will be both more easily obtainable and more reliable than they would be if this reasoning were left to the humans who design or deploy them.

We will proceed on the assumption that one can use some of the same computational tools in computational meta-ethics as have been used in computational metaphysics (Fitelson and Zalta 2007) and in meta-reasoning about different systems of modal logic (Rabe et al. 2009). We make this assumption because we have little reason to believe that meta-ethics is essentially different from (let alone more difficult than) metaphysics or metamathematics.

Design of a Meta-Ethical Robot

Nine Modules

For the purpose of illustration, we envisage a robot with nine modules:

1. Seven non-deontic logical modules.
2. One deontic module.

3. One meta-logical module.

The seven non-deontic modules have the following functions.

1. Module $\mathfrak{S}^{\circ}_{\mathrm{T}}$ implements system $\mathfrak{S}^{\circ}_{\mathrm{T}}$, which we define as *the weakest traditionally strict classical logic closed under strict modus ponens*. $\mathfrak{S}^{\circ}_{\mathrm{T}}$ has the same language as the propositional calculus, except that there is an additional unary connective *square*, read as "necessarily," and the definitions $A \rightarrow B \overset{\mathrm{df}}{=} \Box(A \supset B)$ and $A \leftrightarrow B \overset{\mathrm{df}}{=} (A \rightarrow B) \,\&\, (B \rightarrow A)$, where → is strict implication, ⊃ classical material implication, and ↔ strict equivalence. The axioms and rules of inference are as follows (Chellas and Segerberg 1996):

   PL The set of all tautologies.
   □PL $\{\Box A : A \in \mathrm{PL}\}$.
   US Uniform substitution.
   RRSE$_{\mathrm{T}}$ From $A \leftrightarrow B$ and F to infer $F^{A/B}$, where $F^{A/B}$ is the result of replacing one occurrence of A in F by B.
   MP From A and $A\supset B$ to infer B.
   SMP From A and $A \rightarrow B$ to infer B.
   All Lewis systems of strict implication (as studied in Zeman 1973, for example) are traditionally strict classical logics closed under strict modus ponens. Some of these systems and the relations between them are as follows:

$$\mathfrak{S}^{\circ}_{\mathrm{T}} \subset \mathbf{S0.9}^{\circ} \subset \left\{ \begin{matrix} \mathbf{S0.9} \\ \mathbf{S1}^{\circ} \end{matrix} \right\} \subset \mathbf{S1} \subset \mathbf{S2} \subset \mathbf{S3} \subset \mathbf{S4} \subset \mathbf{S5}.$$

2. Module $\text{Ł}_{\aleph_0}$ implements *Łukasiewicz's infinite-valued logic* $\text{Ł}_{\aleph_0}$, which has the following axioms and rules (Malinowski 2001):

1 $A \rightarrow (B \rightarrow A)$
2 $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$
3 $((A \rightarrow B) \rightarrow B) \rightarrow ((B \rightarrow A) \rightarrow A)$
4 $(\sim A \rightarrow \ \sim B) \rightarrow (B \rightarrow A)$
5 $((A \rightarrow B) \rightarrow (B \rightarrow A)) \rightarrow (B \rightarrow A)$
6 From A and $A \rightarrow B$ to infer B.

   Definitions: $A \vee B \overset{\mathrm{df}}{=} (A \rightarrow B) \rightarrow B$, $A \,\&B \overset{\mathrm{df}}{=} \sim(\sim A \vee \sim B)$, $A \leftrightarrow B \overset{\mathrm{df}}{=} (A \rightarrow B) \ \&(B \rightarrow A)$. $\text{Ł}_{\aleph_0}$ is the weakest of all Łukasiewicz systems, in the sense that A is a theorem of $\text{Ł}_{\aleph_0}$ if and only if A is a theorem of all finite-valued Łukasiewicz calculi $\text{Ł}_n, n \geq 2, n \in \mathbb{N}$. Formally, if $Th(S)$ is the set of theorems of S, then $Th(\text{Ł}_{\aleph_0}) = \bigcap\{Th(\text{Ł}_n) : n \geq 2, n \in \mathbb{N}\}$ (Malinowski 2001). These systems are nowadays popular in the field of *fuzzy logic*.

3. Module **H** implements *Heyting's system of intuitionistic logic* ***H***, which has the following axioms and rules (Dalen 2001):

1 $A \rightarrow (B \rightarrow A)$
2 $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

3 $(A \& B) \rightarrow A$

4 $(A \& B) \rightarrow B$

5 $(A \rightarrow (B \rightarrow (A \& B))$

6 $A \rightarrow (A \vee B)$

7 $B \rightarrow (A \vee B)$

8 $(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \vee B) \rightarrow C))$

9 $A \rightarrow ({\sim}A \rightarrow A)$

10 $(A \rightarrow B) \rightarrow ((A \rightarrow {\sim}B) \rightarrow {\sim}A)$

11 From A and $A \rightarrow B$ to infer B. Intuitionistic logic is related to *minimal logic*, which we will not study separately.

4. Module **R** implements *relevant system **R***, which has the following axioms and rules (Dunn and Restall 2002):

1 $A \rightarrow A$

2 $(A \rightarrow B) \rightarrow ((C \rightarrow A) \rightarrow (C \rightarrow B))$

3 $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

4 $A \rightarrow ((A \rightarrow B) \rightarrow B)$

5 $(A \& B) \rightarrow A$

6 $(A \& B) \rightarrow B$

7 $((A \rightarrow B) \& (A \rightarrow C)) \rightarrow (A \rightarrow (B \& C))$

8 $A \rightarrow (A \vee B)$

9 $B \rightarrow (A \vee B)$

10 $((A \rightarrow C) \& (B \rightarrow C)) \rightarrow ((A \vee B) \rightarrow C)$

11 $(A \& (B \vee C)) \rightarrow ((A \& B) \vee (A \& C))$

12 $(A \rightarrow {\sim}B) \rightarrow (B \rightarrow {\sim}A)$

13 ${\sim}{\sim}A \rightarrow A$

14 From A and $A \rightarrow B$ to infer B.

15 From A and B to infer $A \& B$. Relevance logic is a predecessor of *linear logic*, which we will not study separately.

5. Module **RM** implements *relevant system **RM***, which is defined as **R** plus axiom $A \rightarrow (A \rightarrow A)$.
6. Module **RM3** implements *relevant system **RM3***, which is defined as **R** plus axioms $({\sim}A \& B) \rightarrow (A \rightarrow B)$ and $A \vee (A \rightarrow B)$ (Dunn and Restall 2002).
7. Module **KR** implements *relevant system **KR***, which is defined as **R** plus axiom $(A \& {\sim}A) \rightarrow B$ (Dunn and Restall 2002). Systems **R**, **RM**, **RM3** and **KR** are related as follows: $\mathbf{R} \subset \mathbf{RM} \subset \mathbf{RM3}$, $\mathbf{R} \subset \mathbf{KR}$, $\mathbf{RM} \not\subseteq \mathbf{KR}$, $\mathbf{RM3} \not\subseteq \mathbf{KR}$, $\mathbf{KR} \not\subseteq \mathbf{RM}$, $\mathbf{KR} \not\subseteq \mathbf{RM3}$. System **KR** is famous for being the first relevance logic that was shown to be undecidable.

The deontic module Mally embodies the principles of *Mally's deontic logic*, the very first system of deontic logic, originally published in 1926 (Lokhorst 2008). We use the following symbolic language:

- $O\,A$: it is obligatory that A.
- u: that which is unconditionally obligatory.

- $\circledR_A B \overset{\text{df}}{=} A \rightarrow OB$: A requires B.

Mally's deontic principles are as follows:

1. $(\circledR_A B \,\&\, (B \rightarrow C)) \rightarrow \circledR_A C$
2. $(\circledR_A B \,\&\, \circledR_A C) \rightarrow \circledR_A (B \,\&\, C)$
3. $\circledR_A B \leftrightarrow O(A \rightarrow B)$
4. Ou
5. $\sim \circledR_u \sim u$

Similar postulates have been proposed by others (Anderson 1967, Castañeda 1981). Mally identified → with material implication, but we let its logical properties depend on the context: → is identical with strict implication in the context of $\mathfrak{S}^{\circ}_{\mathrm{T}}$, with Łukasiewicz implication in the context of $Ł_{\aleph_0}$, with intuitionistic implication in the context of **H**, and with relevant implication in the context of **R**, **RM**, **RM3** and **KR**.

Finally, the meta-logical module Meta is the *meta-ethical monitor*. It selects suitable logical modules according to the following criteria:

1. Formula **T1** $A \rightarrow OA$ (if A is the case, then A is obligatory) is *undesirable*.
2. Formula **T2** $OA \rightarrow A$ (if A is obligatory, then A is the case) is *undesirable*.
3. Formula **T3** $O(A \vee B) \rightarrow (OA \vee OB)$ (if it is obligatory that either A or B, then either A is obligatory or B is obligatory) is *undesirable*.
4. Formula **D** $OA \rightarrow \sim O \sim A$ (if A is obligatory, then the negation of A is not obligatory) is *desirable*.

We owe these criteria both to Mally himself and to some of this commentators (see Lokhorst 2008 and Lokhorst 2011).

The nine modules we have mentioned embody insights that were available around 1930. We confine our attention to these ideas because logicians have put forward so many proposals since then that we cannot possibly take all of them into account.

Flexibility of Architecture

The architecture of the robotic reasoning system is *flexible* (modifiable).

- The non-deontic modules can be turned on and off.
- The deontic and meta-ethical modules are fixed.

Modus Operandi

In order to evaluate the logical modules, module Meta is equipped with theorem provers and model generators (which produce counterexamples to formulas). These programs work either concurrently on one processor or in parallel on multiple processors. If the logic under examination is decidable, then given a formula A, either a theorem prover or a model generator will produce a result (if the programs work as advertised).

```
#!/bin/sh
rm -f m.ok p.ok
# remove m.ok and p.ok if they exist; do not prompt
magic -b magic.in >m.out 2>m.err && touch m.ok &
# run magic, redirect outputs, create m.ok when terminated,
# do this in the background
prover9 -f prover9.in >p.out 2>p.err && touch p.ok &
# run prover9, redirect outputs, create p.ok when terminated,
# do this in the background
until [ -f m.ok ] || [ -f p.ok ] ; do sleep 1 ; done
# look once a second if m.ok or p.ok have been created
[ -f m.ok ] && echo refuted ; [ -f p.ok ] && echo proven
# show if m.ok and p.ok have been created and what this means
```

Fig. 1 Simple shell script to execute `MaGIC` and `Prover9` concurrently

The `Unix` shell script displayed in Fig. 1 illustrates this process. The countermodel generator `MaGIC` (Slaney 2008) and the theorem prover `Prover9` (McCune 2008) run concurrently in the background (as indicated by the ampersand & at the end of the respective command lines) until one of them terminates. The value of the argument of `sleep` is irrelevant because only one of the two programs can terminate if the inputs to the programs are equivalent, the logical systems are consistent and the programs work correctly.

The scores for the four reference formulas **T1**, **T2**, **T3** and **D** determine whether module Meta accepts or rejects the logic under examination. Module Meta accepts the logic as soon as each of the undesirable formulas (**T1**, **T2** and **T3**) is shown to be invalid by some countermodel generator and the desirable formula **D** is shown to be derivable by some theorem prover. Module Meta rejects the logic it as soon as some undesirable formula (**T1**, **T2** or **T3**) is shown to be derivable by some theorem prover or the desirable formula **D** is shown to be invalid by some countermodel generator. Module Meta remains in a state of indecision as long as the logic under examination has not been accepted or rejected. This state of indecision may last forever in the case of **R** and **KR** because these systems are undecidable. This would be fatal for a robot on the battlefield or in a hospital, but we have not encountered this situation in the cases we are examining.

Meta-Ethical Reasoning Process

In the example that we are going to describe, module Meta contains one theorem prover, namely `Prover9` (McCune 2008), and two countermodel generators, namely `Mace4` (McCune 2008) and `MaGIC` (Slaney 2008). The latter program is especially suitable in the case of relevant systems, but it was useful for the refutation of **T3** in **H** as well.

The meta-ethical reasoning process proceeds as follows. The robot starts with modules $\mathfrak{S}^{\circ}_{\mathrm{T}}$ and Mally. It connects these to module Meta (Fig. 2).Theorem prover `Prover9` quickly derives theorems **T1**, **T2** and **T3** and **D**. The countermodel generators produce no result. From this, the robot concludes that $\mathfrak{S}^{\circ}_{\mathrm{T}}$ plus Mally is unacceptable in the light of its meta-ethical standards (see Lokhorst 2010 and Lokhorst 2011 for the details). This concludes the case of $\mathfrak{S}^{\circ}_{\mathrm{T}}$. The case of $\text{Ł}_{\aleph_0}$ is

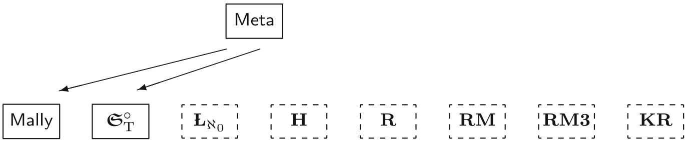

Fig. 2 Meta-ethical reasoning process, stage 1: probing Mally and $\mathfrak{S}^{\circ}_{\mathrm{T}}$

more difficult. It takes `Prover9` several hours to prove that **T1** and **T2** are theorems of $\text{Ł}_{\aleph_0}$ plus Mally's axioms; the derivations of these formulas are about a hundred lines long. The derivations of **T1** and **T2** are presented in the Appendix, if only to demonstrate that it is not advisable to do this kind of reasoning without computer assistance. The remaining cases (**H** and the relevant systems) are relatively easy, both for computers and humans (even though the refutation of **T3** in **H** required some human ingenuity, as described in Lokhorst 2011).

The results for the seven systems and the four bench-mark formulas are summed up in the following table, in which + indicates derivability and − invalidity:

Formula	D1–D5 plus						
	$\mathfrak{S}^{\circ}_{\mathrm{T}}$	$\text{Ł}_{\aleph_0}$	**H**	**R**	**RM**	**RM3**	**KR**
T1 $A \rightarrow OA$	+	+	+	−	−	−	−
T2 $OA \rightarrow A$	+	+	−	−	−	+	−
T3 $O(A \vee B) \rightarrow (OA \vee OB)$	+	+	−	−	+	+	−
D $OA \rightarrow {\sim}O{\sim}A$	+	+	+	−	−	+	+

The table shows that only **KR** is acceptable in view of the meta-ethical standards employed by the robot.

Conclusion

We have shown that computational meta-ethics is to some extent feasible, using current, off-the-shelf, generic, freely available open-source software.

The above design of a meta-ethical robot is nothing but a "proof of concept." Further work is needed. We mention the following themes.

- Different logical systems, both deontic and non-deontic.
- Different theorem-provers and model-generators, for example the award-winning `Vampire` and `Paradox` (Rabe et al. 2009).
- Different meta-ethical criteria, for example, acceptability and unacceptability of alternative deontic principles. Freedom from "is-ought fallacies" would be an example. Is-ought fallacies are formulas of the form $A \rightarrow OB$ or $A \rightarrow {\sim}OB$, where A and B contain no occurrences of O and u. It can be proven that **R** plus Mally's axioms is the only system on our list that avoids such fallacies, but we doubt whether the computer is clever enough to see this.

- Different ways of reasoning about (moral) reasoning, for example in terms of computational complexity and computational tractability.
- More expressive languages, for example languages with perception, knowledge, action, multiple agents, strategies, intention and time (see Hoven and Lokhorst 2002 and Lokhorst and Hoven 2011 for further discussion). In this context, it is interesting to know that large parts of multimodal correspondence theory have recently been mechanized, which makes it easier to study the *interactions* of modalities such as belief, seeing to it that, possibility, obligation and temporal notions (Georgiev et al. 2006).
- On the fringe of logic and beyond logic: probabilistic reasoning, moral belief revision, the moral frame problem, non-monotonic reasoning and non-inferential moral judgment, for example case-based judgment and pattern recognition with neural networks (see Gärdenfors 2005 for more about these topics).

In other words, the design of a fully fledged meta-ethical robot is still a long way off. However, developments in the recent past suggest at least one direction in which we can set out on the long road that lies ahead.

Appendix

Proofs of **T1** and **T2** in system $\text{Ł}_{\aleph_0}$. The derivations were generated by `Prover9` (McCune 2008). The symbols can be read as follows:

Meta-level		*Object-level*	
`P`	provable	`a`	or
`-`	not	`b`	the unconditionally obligatory
`|`	or	`i`	implies
`=`	$\stackrel{\text{df}}{=}$	`k`	and
`$F`	absurd	`o`	obligatory
		`n`	not

Derivation of **T1**: $A \rightarrow OA$.

```
1 P(i(x,o(x))) # label(non_clause). [goal].
2 P(i(x,i(y,x))). [assumption].
3 P(i(i(x,y),i(i(y,z),i(x,z)))). [assumption].
4 P(i(i(i(x,y),y),i(i(y,x),x))). [assumption].
5 P(i(i(i(x,y),i(y,x)),i(y,x))). [assumption].
6 P(i(i(n(x),n(y)),i(y,x))). [assumption].
7 a(x,y) = i(i(x,y),y). [assumption].
```

```
8 k(x,y) = n(a(n(x),n(y))). [assumption].
9 k(x,y) = n(i(i(n(x),n(y)),n(y))). [8,7].
10 P(i(k(i(x,o(y)),i(y,z)),i(x,o(z)))). [assumption].
11 P(i(n(i(i(n(i(x,o(y))),n(i(y,z))),n(i(y,z)))),i(x,o(z)))).
    [10,9].
12 P(i(i(x,o(y)),o(i(x,y)))). [assumption].
13 P(o(b)). [assumption].
14 P(n(i(b,o(n(b))))). [assumption].
15 -P(x) | -P(i(x,y)) | P(y). [assumption].
16 -P(i(c1,o(c1))). [1].
17 P(i(x,n(i(b,o(n(b)))))). [15,14,2].
18 P(i(x,o(b))). [15,13,2].
19 P(i(i(i(x,y),z),i(i(n(y),n(x)),z))). [15,6,3].
20 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))). [15,3,3].
21 P(i(i(i(x,y),z),i(y,z))). [15,2,3].
22 P(i(x,i(y,i(z,y)))). [15,2,2].
23 -P(n(i(i(n(i(c1,o(x))),n(i(x,c1))),n(i(x,c1))))).
   [15,11,16].
24 P(i(i(b,o(n(b))),x)). [15,17,6].
25 P(i(i(o(b),x),x)). [15,18,4].
26 P(i(x,i(y,o(b)))). [15,18,2].
27 P(i(i(x,y),i(i(o(b),x),y))). [15,25,3].
28 P(i(i(i(x,o(b)),y),y)). [15,26,4].
29 P(i(i(i(x,i(y,x)),z),z)). [15,22,4].
30 P(i(o(n(b)),b)). [15,24,5].
31 P(i(i(o(n(b)),b),b)). [15,24,4].
32 P(b). [15,30,31].
33 P(i(x,b)). [15,32,2].
34 P(i(i(b,x),x)). [15,33,4].
35 P(i(i(x,y),i(i(b,x),y))). [15,34,3].
36 P(i(i(x,y),i(i(i(z,o(b)),x),y))). [15,28,3].
37 P(i(n(x),i(x,y))). [15,6,21].
38 P(i(x,i(i(x,y),y))). [15,4,21].
39 P(i(i(n(x),n(b)),x)). [15,34,19].
40 P(i(i(i(x,y),z),i(n(x),z))). [15,37,3].
41 P(i(i(i(i(x,y),y),z),i(x,z))). [15,38,3].
42 P(i(i(i(i(n(x),n(b)),x),y),y)). [15,39,38].
43 P(i(n(b),x)). [15,39,21].
44 P(i(i(n(n(b)),n(n(x))),x)). [15,39,19].
45 P(i(i(i(n(b),x),y),y)). [15,43,38].
46 P(i(i(x,n(b)),i(x,y))). [15,45,20].
47 P(i(i(b,n(b)),x)). [15,34,46].
48 P(i(i(i(i(b,n(b)),x),y),y)). [15,47,38].
49 P(i(i(x,y),i(x,i(z,y)))). [15,29,20].
50 P(i(i(b,x),i(i(x,y),y))). [15,38,35].
51 P(i(n(n(x)),x)). [15,44,21].
```

```
52 P(i(x,n(n(x)))). [15,51,6].
53 P(i(i(x,y),i(n(n(x)),y))). [15,51,3].
54 P(i(i(n(n(x)),y),i(x,y))). [15,52,3].
55 P(i(i(o(b),i(b,x)),i(i(x,y),y))). [15,50,27].
56 P(i(x,i(n(x),y))). [15,37,54].
57 P(i(i(i(n(x),y),z),i(x,z))). [15,56,3].
58 P(i(n(x),o(i(x,y)))). [15,12,40].
59 P(i(x,o(i(n(x),y)))). [15,58,54].
60 P(i(n(i(x,o(i(n(x),y)))),z)). [15,59,56].
61 P(i(i(x,i(b,n(b))),i(x,y))). [15,48,20].
62 P(i(i(b,x),i(i(x,n(b)),y))). [15,61,20].
63 P(i(x,i(i(x,n(b)),y))). [15,62,21].
64 P(i(n(n(x)),i(i(x,n(b)),y))). [15,63,53].
65 P(i(i(x,y),i(i(z,x),i(z,y)))). [15,20,41].
66 P(i(i(x,i(y,z)),i(y,i(x,z)))). [15,41,20].
67 P(i(i(x,i(n(y),n(b))),i(x,y))). [15,42,20].
68 P(i(i(i(x,o(b)),y),i(i(y,z),z))). [15,38,36].
69 P(i(i(x,n(b)),i(n(n(x)),y))). [15,64,66].
70 P(i(i(x,n(b)),n(x))). [15,69,67].
71 P(i(i(i(x,y),n(b)),x)). [15,40,67].
72 P(i(i(x,i(y,n(b))),i(x,n(y)))). [15,70,65].
73 P(i(x,i(i(i(y,z),n(b)),y))). [15,71,2].
74 P(i(i(x,y),i(i(y,n(b)),n(x)))). [15,72,20].
75 P(i(i(x,n(b)),i(i(y,x),n(y)))). [15,74,66].
76 P(i(x,i(i(y,n(x)),n(y)))). [15,75,57].
77 P(i(n(x),i(i(y,x),n(y)))). [15,75,40].
78 P(i(x,i(y,i(i(z,n(x)),n(z))))). [15,76,49].
79 P(i(i(x,y),i(n(y),n(x)))). [15,77,66].
80 P(i(x,i(n(y),n(i(x,y))))). [15,79,41].
81 P(i(n(x),i(y,n(i(y,x))))). [15,80,66].
82 P(i(i(i(i(x,n(o(b))),n(x)),y),y)). [15,78,55].
83 P(i(i(i(i(i(x,y),n(b)),x),z),z)). [15,73,68].
84 P(i(i(n(x),y),i(i(x,n(o(b))),y))). [15,3,82].
85 P(i(i(x,i(i(y,z),n(b))),i(x,y))). [15,65,83].
86 P(i(i(x,n(o(b))),i(y,n(i(y,x))))). [15,81,84].
87 P(i(i(i(x,y),z),i(i(z,n(b)),x))). [15,85,20].
88 P(i(x,n(i(x,n(i(y,o(i(n(y),z)))))))). [15,60,86].
89 P(i(i(x,n(b)),i(i(i(y,z),x),y))). [15,87,66].
90 P(i(x,i(i(i(y,z),n(x)),y))). [15,89,57].
91 P(i(i(i(x,y),n(z)),i(z,x))). [15,90,66].
92 P(i(n(i(x,y)),n(i(i(y,z),n(x))))). [15,91,79].
93 -P(n(i(i(x,c1),n(i(c1,o(x)))))). [15,92,23].
94 P(i(i(n(x),n(i(y,o(i(n(y),z))))),x)). [15,88,6].
95 -P(i(i(n(c1),x),c1)). [15,88,93].
96 $F. [95,94].
```

Derivation of **T2**: $OA \rightarrow A$.

```
1 P(i(o(x),x)) # label(non_clause). [goal].
2 P(i(x,i(y,x))). [assumption].
3 P(i(i(x,y),i(i(y,z),i(x,z)))). [assumption].
4 P(i(i(i(x,y),y),i(i(y,x),x))). [assumption].
5 P(i(i(i(x,y),i(y,x)),i(y,x))). [assumption].
6 P(i(i(n(x),n(y)),i(y,x))). [assumption].
7 a(x,y) = i(i(x,y),y). [assumption].
8 k(x,y) = n(a(n(x),n(y))). [assumption].
9 k(x,y) = n(i(i(n(x),n(y)),n(y))). [8,7].
10 P(i(k(i(x,o(y)),i(y,z)),i(x,o(z)))). [assumption].
11 P(i(n(i(i(n(i(x,o(y))),n(i(y,z))),n(i(y,z)))),i(x,o(z)))).
   [10,9].
12 P(i(i(x,o(y)),o(i(x,y)))). [assumption].
13 P(i(o(i(x,y)),i(x,o(y)))). [assumption].
14 P(o(b)). [assumption].
15 P(n(i(b,o(n(b))))). [assumption].
16 -P(x) | -P(i(x,y)) | P(y). [assumption].
17 P(i(x,o(x))). [assumption]. # proven above
18 -P(i(o(c1),c1)). [1].
19 P(i(x,n(i(b,o(n(b)))))). [16,15,2].
20 P(i(i(i(x,o(y)),z),i(o(i(x,y)),z))). [16,13,3].
21 P(i(i(i(x,o(y)),z),i(n(i(i(n(i(x,o(u))),n(i(u,y))),
   n(i(u,y)))),z))). [16,11,3].
22 P(i(i(i(x,y),z),i(i(n(y),n(x)),z))). [16,6,3].
23 P(i(i(i(x,y),z),i(i(i(y,x),i(x,y)),z))). [16,5,3].
24 P(i(i(i(i(x,y),y),z),i(i(i(y,x),x),z))). [16,4,3].
25 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))). [16,3,3].
26 P(i(i(i(x,y),z),i(y,z))). [16,2,3].
27 P(o(o(b))). [16,14,17].
28 P(i(i(o(x),y),i(x,y))). [16,17,3].
29 P(i(i(b,o(n(b))),x)). [16,19,6].
30 P(i(x,o(o(b)))). [16,27,2].
31 P(i(o(i(x,y)),o(i(x,o(y))))). [16,17,20].
32 P(i(i(o(o(b)),x),x)). [16,30,4].
33 P(i(n(i(i(n(i(b,o(x))),n(i(x,n(b)))),n(i(x,n(b))))),y)).
   [16,29,21].
34 P(i(o(n(b)),b)). [16,29,5].
35 P(i(i(o(n(b)),b),b)). [16,29,4].
36 P(i(i(x,y),i(i(o(o(b)),x),y))). [16,32,3].
37 P(i(i(n(x),n(y)),o(i(y,x)))). [16,17,22].
38 P(i(i(n(x),n(y)),i(i(x,z),i(y,z)))). [16,3,22].
39 P(i(i(i(x,y),i(y,x)),i(i(x,z),i(y,z)))). [16,3,23].
40 P(i(i(i(o(x),y),y),i(o(i(y,x)),o(x)))). [16,20,24].
41 P(b). [16,34,35].
```

```
42 P(i(x,b)). [16,41,2].
43 P(i(i(b,x),x)). [16,42,4].
44 P(i(i(n(x),n(b)),x)). [16,43,22].
45 P(o(i(i(b,x),x))). [16,43,17].
46 P(i(i(x,y),i(i(b,x),y))). [16,43,3].
47 P(i(i(b,x),o(x))). [16,45,13].
48 P(o(i(i(b,x),o(x)))). [16,47,17].
49 P(i(i(o(x),y),i(i(b,x),y))). [16,47,3].
50 P(i(x,x)). [16,43,26].
51 P(i(o(n(b)),x)). [16,29,26].
52 P(i(n(x),i(x,y))). [16,6,26].
53 P(i(x,i(i(x,y),y))). [16,4,26].
54 P(i(x,i(y,i(z,x)))). [16,2,26].
55 P(i(o(i(i(x,y),z)),i(y,o(z)))). [16,26,20].
56 P(o(i(o(x),x))). [16,50,12].
57 P(i(i(i(x,y),z),i(n(x),z))). [16,52,3].
58 P(i(i(n(n(b)),n(n(x))),x)). [16,44,22].
59 P(i(i(b,x),o(o(x)))). [16,48,13].
60 P(i(i(i(o(n(b)),x),y),y)). [16,51,53].
61 P(i(x,i(y,i(i(y,z),z)))). [16,53,2].
62 P(i(x,i(y,o(i(o(z),z))))). [16,56,54].
63 P(i(i(x,y),i(x,o(y)))). [16,13,28].
64 P(i(n(n(x)),x)). [16,58,26].
65 P(n(n(b))). [16,64,44].
66 P(i(x,n(n(x)))). [16,64,6].
67 P(i(i(x,y),i(n(n(x)),y))). [16,64,3].
68 P(i(x,o(n(n(x))))). [16,66,63].
69 P(i(i(i(x,n(n(x))),y),y)). [16,66,53].
70 P(i(x,n(n(i(y,x))))). [16,66,26].
71 P(i(i(n(n(x)),y),i(x,y))). [16,66,3].
72 P(i(x,o(n(n(o(x)))))). [16,68,28].
73 P(i(n(i(x,n(y))),y)). [16,70,6].
74 P(i(i(x,o(n(b))),i(x,y))). [16,60,25].
75 P(i(n(n(x)),n(n(i(y,x))))). [16,70,67].
76 P(i(i(x,y),i(x,n(n(y))))). [16,69,25].
77 P(i(x,i(n(x),y))). [16,52,71].
78 P(i(i(b,x),i(n(x),y))). [16,77,46].
79 P(i(n(i(x,y)),n(y))). [16,75,6].
80 P(i(n(i(x,y)),n(n(n(y))))). [16,79,76].
81 P(i(x,n(n(i(i(x,y),y))))). [16,53,76].
82 P(i(n(x),n(n(i(x,y))))). [16,52,76].
83 P(i(n(i(x,y)),x)). [16,82,6].
84 -P(n(i(i(o(c1),c1),x))). [16,83,18].
85 P(i(n(i(i(n(x),y),y)),x)). [16,81,6].
86 P(o(i(x,i(i(n(n(x)),y),y)))). [16,85,37].
87 P(i(i(b,x),o(n(n(o(o(x))))))). [16,72,49].
```

```
88 P(i(n(x),i(i(i(x,y),z),z))). [16,53,57].
89 P(i(x,o(i(i(n(n(x)),y),y)))). [16,86,13].
90 P(o(i(i(b,x),n(n(o(o(x))))))). [16,87,12].
91 P(o(i(i(n(n(n(n(b)))),x),x))). [16,65,89].
92 P(o(i(i(b,x),o(n(n(o(o(x)))))))). [16,90,31].
93 P(i(i(n(n(n(n(b)))),x),o(x))). [16,91,13].
94 P(o(i(i(i(n(n(n(b))),x),y),y))). [16,88,93].
95 P(i(i(o(o(b)),x),i(n(x),y))). [16,77,36].
96 P(i(n(i(x,o(i(o(y),y)))),z)). [16,62,95].
97 P(i(i(i(x,y),z),i(y,o(z)))). [16,55,28].
98 P(i(i(b,x),o(o(n(n(o(o(x)))))))). [16,92,13].
99 P(i(i(i(n(n(n(b))),x),y),o(y))). [16,94,13].
100 P(i(x,o(o(o(n(n(o(o(x))))))))). [16,98,97].
101 P(i(n(o(o(o(n(n(o(o(b)))))))),x)). [16,100,78].
102 P(i(i(x,n(n(n(b)))),o(i(x,y)))). [16,99,25].
103 P(o(i(n(o(o(o(n(n(o(o(b)))))))),x))). [16,101,102].
104 P(i(i(i(x,y),z),i(n(n(y)),z))). [16,80,38].
105 P(i(i(i(x,n(n(y))),z),i(y,z))). [16,73,38].
106 P(i(i(i(i(x,y),y),z),i(x,z))). [16,61,39].
107 P(i(x,i(i(x,o(n(b))),y))). [16,74,106].
108 P(i(i(x,y),i(i(z,x),i(z,y)))). [16,25,106].
109 P(i(x,i(i(n(y),n(x)),y))). [16,22,106].
110 P(i(i(x,i(y,z)),i(y,i(x,z)))). [16,106,25].
111 P(i(n(n(x)),i(i(x,o(n(b))),y))). [16,107,67].
112 P(i(i(x,i(b,y)),i(x,o(o(y))))). [16,59,108].
113 P(i(i(x,i(n(y),n(b))),i(x,y))). [16,44,108].
114 P(i(i(x,o(n(b))),i(n(n(x)),y))). [16,111,110].
115 P(i(i(b,x),i(i(x,y),o(o(y))))). [16,112,25].
116 P(i(i(x,o(n(b))),n(x))). [16,114,113].
117 P(i(o(i(x,n(b))),n(x))). [16,116,20].
118 P(o(i(o(i(x,n(b))),n(x)))). [16,117,17].
119 P(i(n(n(x)),i(o(i(x,y)),o(y)))). [16,40,104].
120 P(i(x,i(i(n(n(x)),y),o(o(y))))). [16,115,105].
121 P(i(o(i(x,y)),i(n(n(x)),o(y)))). [16,119,110].
122 P(i(i(n(n(x)),y),i(x,o(o(y))))). [16,120,110].
123 P(i(n(n(o(i(x,n(b))))),o(n(x)))). [16,118,121].
124 P(i(o(i(x,n(b))),o(o(o(n(x)))))). [16,123,122].
125 P(o(o(o(n(n(o(o(o(n(n(o(o(b))))))))))))). [16,103,124].
126 P(i(i(n(x),n(o(o(o(n(n(o(o(o(n(n(o(o(b))))))))))))),x)).
    [16,125,109].
127 P(i(i(n(i(b,o(x))),n(i(x,n(b)))),n(i(x,n(b))))).
    [16,33,126].
128 P(n(i(i(o(x),x),n(b)))). [16,96,127].
129 $F. [128,84].
```

References

Anderson, A. R. (1967). Some nasty problems in the formal logic of ethics. *Noûs, 1*, 345–360.

Arkin, R. C. (2009). *Governing lethal behavior in autonomous robots*. Boca Raton, Fl.: Chapman & Hall.

Arkoudas, K., Bringsjord, S., & Bello, P. (2005). Toward ethical robots via mechanized deontic logic. In *Machine ethics: Papers from the AAAI fall symposium*, Technical Report FS–05–06. Menlo Park, Cal.: AAAI Press.

Bringsjord, S., Arkoudas, K., & Bello, P. (2006). Toward a general logicist methodology for engineering ethically correct robots. *IEEE Intelligent Systems, 21*(4), 38–44.

Carnap, R. (1937). *The logical syntax of language*. London: Routledge and Kegan Paul.

Castañeda, H.-N. (1981). The paradoxes of deontic logic: The simplest solution to all of them in one fell swoop. In R. Hilpinen (Ed.) *New studies in deontic logic*. Dordrecht: Reidel.

Chellas, B. F., & Segerberg, K. (1996). Modal logics in the vicinity of **S1**. *Notre Dame Journal of Formal Logic, 37*, 1–24.

Dunn, J. M., & Restall, G. (2002). Relevance logic. In D. Gabbay & F. Guenthner (Eds.), *Handbook of philosophical logic* (2nd Edn., Vol. 6). Dordrecht: Kluwer.

Fitelson, B., & Zalta, E. N. (2007). Steps toward a computational metaphysics. *Journal of Philosophical Logic, 36*, 227–247.

Gärdenfors, P. (2005). *The dynamics of thought*. Dordrecht: Springer.

Georgiev, D., Tinchev, T., & Vakarelov, D. (2006). `Sqema` (an algorithm for computing first-order correspondences in modal logic) version 0.9.8, September 2006. http://www.fmi.uni-sofia.bg/fmi/logic/sqema/.

Lokhorst, G. J. C. (2008). Mally's deontic logic. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy*. plato.stanford.edu/entries/mally-deontic/.

Lokhorst, G. J. C. (2010). Where did Mally go wrong? *Lecture Notes in Artificial Intelligence, 6181*, 247–258.

Lokhorst, G. J. C. (2011). Where did Mally go wrong? *Journal of Applied Logic* (accepted).

Lokhorst, G. J. C., & van den Hoven, M. J. (2011). Responsibility for robots. In P. Lin, K. Abney & G. Bekey (Eds.) *Robot ethics: The ethical and social implications of robotics*. Cambridge, Mass.: MIT Press. (Forthcoming in 2011).

Malinowski, G. (2001). Many-valued logics. In L. Goble (Ed.), *The Blackwell guide to philosophical logic*. Oxford: Blackwell.

McCune, W. (2008). `Prover9` and `Mace4` version 2008-11A, November 2008. http://www.cs.unm.edu/~mccune/prover9/.

Moor, J. H. (2006). The nature, importance, and difficulty of machine ethics. *IEEE Intelligent Systems, 21*(4), 18–21.

Rabe, F., Pudlák, P., Sutcliffe, G., & Shen, W. (2009). Solving the $100 modal logic challenge. *Journal of Applied Logic, 7*, 113–130.

Slaney, J. K. (2008). `MaGIC` (Matrix Generator for Implication Connectives) version 2.2.1, November 2008. users.rsise.anu.edu.au/~jks/magic.html.

Van Dalen, D. (2001). Intuitionistic logic. In L. Goble (Ed.), *The Blackwell guide to philosophical logic*. Oxford: Blackwell.

Van den Hoven, M. J., & Lokhorst, G. J. C. (2002). Deontic logic and computer-supported computer ethics. *Metaphilosophy*, *33*, 376–386. Reprinted in J. H. Moor & T. W. Bynum (Eds.) (2002). *CyberPhilosophy*. Oxford: Blackwell.

Wallach, W., & Allen, C. (2008). *Moral machines: Teaching robots right from wrong*. Oxford: Oxford University Press.

Zeman, J. J. (1973). *Modal logic: The Lewis-Modal systems*. Oxford: The Clarendon Press. http://www.clas.ufl.edu/users/jzeman/modallogic/.