Automated Reasoning for Robot Ethics*

Ulrich Furbach¹, Claudia Schon¹ and Frieder Stolzenburg²

¹ Universität Koblenz-Landau, {uli,schon}@uni-koblenz.de
² Harz University of Applied Sciences, fstolzenburg@hs-harz.de

Abstract. Deontic logic is a very well researched branch of mathematical logic and philosophy. Various kinds of deontic logics are considered for different application domains like argumentation theory, legal reasoning, and acts in multi-agent systems. In this paper, we show how standard deontic logic can be used to model ethical codes for multi-agent systems. Furthermore we show how Hyper, a high performance theorem prover, can be used to prove properties of these ethical codes.

1 Introduction

Deontic logic is a very well researched branch of mathematical logic and philosophy. Various kinds of deontic logics are considered for different application domains like argumentation theory, legal reasoning, and acts in multi-agent systems [8].

This paper applies automated reasoning in standard deontic logic (SDL) to a problem in multi-agent systems. For this, we follow the example from [5] where reasoning with ethical codes is presented. Instead of implementing a reasoning system for deontic logic directly, we rely on the first order reasoning system Hyper [14], which uses the hypertableau calculus from [3]. This has the advantage that reasoning in deontic logic can be embedded easily into other applications for which Hyper is already an inference system (e.g. the LogAnswer system [6]).

SDL corresponds to the modal logic $K$ with a seriality axiom. Due to the fact that modal logic $K$ is (more or less) a notational variant of description logic $\mathcal{ALC}$, deontic logic can be translated into it [7]. Since Hyper offers a decision procedure for various description logics, it can be employed for reasoning in SDL.

In Section 2 we shortly depict the reasoning framework, and in Section 3 we show how to model our small example. This hopefully documents the applicability of reasoning with SDL in multi-agent systems.

2 Automated Reasoning for Standard Deontic Logic

Standard deontic logic (SDL) is obtained from the well-known modal logic $K$ by adding the seriality axiom $D$: $\Box P \rightarrow \Diamond P$. In this logic, the $\Box$-operator is

* This is a short version of [7]. It is supported by the DFG grants FU 263/15-1 and STO 421/5-1 'Ratiolog'.
interpreted as ‘it is obligatory that’ and the ♦ as ‘it is permitted that’. The ♦-operator can be defined by ♦P ≡ ¬□¬P. The seriality axiom in SDL states that, if a formula has to hold in all reachable worlds, then there exists such a world. With the deontic reading of □ and ♦ this means: Whenever the formula P ought to be, then there exists a world where it holds. In consequence, there is always a world, which is ideal in the sense that all the norms formulated by ‘the ought to be’-operator hold.

Deontic logic is the logic of choice when formalizing knowledge about norms like the representation of legal knowledge or ethical codes for agents. However, there are only few automated theorem provers specially dedicated for deontic logic and used by deontic logicians (see [12]). Nonetheless, numerous approaches to translate modal logics into (decidable fragments of) first-order predicate logics are stated in the literature. A nice overview including many relevant references is given in [12].

In this paper, we use the first order predicate logic theorem prover Hyper [14] to handle SDL knowledge bases. By using the well-known translation of modal logic into description logic [11] (called ϕ in the sequel) and the fact that Hyper offers a decision procedure for the description logic $SHIQ^d$, we are able to process SDL efficiently. In the following, SDL is translated into $ALC$, which is a subset of $SHIQ^d$.

**Transformation from SDL into $ALC$**

For a normative system consisting of the set of deontic logic formulae $N = \{F_1, \ldots, F_n\}$, the translation $\varphi$ is defined as the conjunctive combination of the translation of all deontic logic formulae in $N$:

$$\varphi(N) = \varphi(F_1) \cap \ldots \cap \varphi(F_n)$$

Note that $\varphi(N)$ does not yet contain the translation of the seriality axiom. As shown in [9], the seriality axiom can be translated into the TBox

$$\mathcal{T} = \{\top \sqsubseteq \exists r. \top\}$$

with $r$ the atomic role introduced by the mapping $\varphi$.

For our application, the result of the translation of a normative system $N$ and the seriality axiom is an $ALC$ knowledge base $\Phi(N) = (\mathcal{T}, \mathcal{A})$, where the TBox $\mathcal{T}$ consists of the translation of the seriality axiom and the ABox $\mathcal{A} = \{\varphi(N)(a)\}$ for a new individual $a$. In description logics, performing a satisfiability test of a concept $C$ w.r.t. a TBox is usually done by adding a new individual $a$ together with the ABox assertion $C(a)$. For the sake of simplicity, we do this construction already during the transformation of $\Phi$ by adding $(\varphi(N))(a)$ to the ABox.

An advantage of the translation of deontic logic formulae into an $ALC$ knowledge base is the existence of a TBox in $ALC$. This makes it possible to add further axioms to the TBox. For example we can add certain norms that we want to be satisfied in all reachable worlds into the TBox.
**Reasoning Tasks**

With the help of Hyper, we can solve the following interesting reasoning tasks:

*Consistency checking of normative systems:* In practice, normative systems can be very large. Therefore it is not easy to see, if a given normative system is consistent. The Hyper theorem prover can be used to check consistency of a normative system $N$.

*Evaluation of normative systems:* Given several normative systems, we use Hyper to find out for which normative system a desired outcome is guaranteed.

**3 An Example from Multi-agent Systems**

In multi-agent systems, there is a challenging area of research, namely the formalization of ‘robot ethics’. It aims at defining formal rules for the behavior of agents and to prove certain properties. As an example consider Asimov’s laws, which aim at regulating the relation between robots and humans. In [3], the authors depict a small example of two surgery robots obeying ethical codes concerning their work. These codes are expressed by means of MADL, which is an extension of standard deontic logic with two operators. In [10], an axiomatization of MADL is given. Further it is asserted, that MADL is not essentially different from standard deontic logic. This is why we use SDL to model the example.

**Formalization in SDL**

In our example, there are two robots $ag1$ and $ag2$ in a hospital. For the sake of simplicity, each robot can perform one specific action: $ag1$ can terminate a person’s life support and $ag2$ can delay the delivery of pain medication. In [5], four different ethical codes $J$, $J^*$, $O$ and $O^*$ are considered:

- “If ethical code $J$ holds, then robot $ag1$ ought to take care that life support is terminated.” This is formalized as:
  $$J \rightarrow \Box act(ag1, \text{term})$$

- “If ethical code $J^*$ holds, then code $J$ holds, and robot $ag2$ ought to take care that the delivery of pain medication is delayed.” This is formalized as:
  $$J^* \rightarrow J \land J^* \rightarrow \Box act(ag2, \text{delay})$$

- “If ethical code $O$ holds, then robot $ag2$ ought to take care that delivery of pain medication is not delayed.” This is formalized as:
  $$O \rightarrow \Box \neg act(ag2, \text{delay})$$
“If ethical code $O^*$ holds, then code $O$ holds, and robot $ag1$ ought to take care that life support is not terminated.” This is formalized as:

$$O^* \rightarrow O \land O^* \rightarrow \Box \neg act(ag1, term)$$

Further we give a slightly modified version of the evaluation of the acts of the robots, as stated in [5], where $(+!!)$ denotes the most and $(−!!)$ the least desired outcome. Note that terms like $(+!!)$ are just propositional atomic formulae here.

$$act(ag1, term) \land act(ag2, delay) \rightarrow (−!!) \quad (1)$$

$$act(ag1, term) \land \neg act(ag2, delay) \rightarrow (−!) \quad (2)$$

$$\neg act(ag1, term) \land act(ag2, delay) \rightarrow (−) \quad (3)$$

$$\neg act(ag1, term) \land \neg act(ag2, delay) \rightarrow (+!!) \quad (4)$$

These formulae evaluate the outcome of the robots’ actions. It makes sense to assume, that this evaluation is effective in all reachable worlds. This is why we add formulae stating that formulae (1)–(4) hold in all reachable worlds. For example, for (1) we add:

$$\Box (act(ag1, term) \land act(ag2, delay) \rightarrow (−!!)) \quad (5)$$

Since our example does not include nested modal operators, the formulae of the form (5) are sufficient to spread the evaluation formulae to all reachable worlds. The normative system $N$ formalizing this example consists of the formalization of the four ethical codes and the formulae for the evaluation of the robots actions.

**Reduction to a Satisfiability Test**

A possible query would be to ask if the most desirable outcome $(+!!)$ will come to pass if ethical code $O^*$ is operative. This query can be translated into a satisfiability test. If $N \land O^* \land \Diamond \neg(+(!!))$ is unsatisfiable, then ethical code $O^*$ ensures outcome $(+!!)$.

Hyper can be used for this satisfiability test. We obtain the desired result namely that (only) ethical code $O^*$ leads to the most desirable behavior $(+!!)$.

**Experiment**

We formalized this example and tested it with the Hyper theorem prover as described above. Since all formalizations are available in $\mathcal{ALC}$, we used the description logic reasoner Pellet [13] to compare the performance with Hyper. It took Pellet 2.548 seconds and Hyper 2.596 seconds to show the unsatisfiability.

Additional experiments can be found in [7]. For the examples we considered, the runtimes of Pellet and Hyper are comparable. Further investigation and comparison with other modal and/or description logic reasoning tools is required and subject of future work.
4 Conclusion

In this paper, we illustrated with the help of an example how to use standard deontic logic to model ethical codes for multi-agent systems. For normative systems described with deontic logic, there is a translation into description logic concepts. These concepts can be checked automatically by automated theorem provers. We used the Hyper theorem prover to prove that a specified desired outcome is guaranteed, if a certain ethical code is operative.

References

1. A. Artosi, P. Cattabriga, and G. Governatori. KED: A deontic theorem prover. In *On Legal Application of Logic Programming, ICLP’94*, pages 60–76, 1994.
2. N. Bassiliades, E. Kontopoulos, G. Governatori, and G. Antoniou. A modal defeasible reasoner of deontic logic for the semantic web. *Int. J. Semant. Web Inf. Syst.*, 7(1):18–43, Jan. 2011.
3. P. Baumgartner, U. Furbach, and I. Niemelä. Hyper tableaux. In J. J. Alferes, L. M. Pereira, and E. Orlowska, editors, *JELIA*, volume 1126 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 1996.
4. M. Bender, B. Pelzer, and C. Schon. System description: E-KRHyper 1.4 - extensions for unique names and description logic. In M. P. Bonacina, editor, *CADE-24*, volume 7898 of *LNCS*, pages 126–134. Springer, 2013.
5. S. Bringsjord, K. Arkoudas, and P. Bello. Toward a general logicist methodology for engineering ethically correct robots. *IEEE Intelligent Systems*, 21(4):38–44, 2006.
6. U. Furbach, I. Glöckner, and B. Pelzer. An application of automated reasoning in natural language question answering. *AI Commun.*, 23(2-3):241–265, 2010.
7. U. Furbach, C. Schon, and F. Stolzenburg. Automated reasoning in deontic logic. In M. N. Murty, X. He, C. R. Rao, and P. Weng, editors, *Multi-disciplinary Trends in Artificial Intelligence - 8th International Workshop, MIWAI 2014, Bangalore, India, December 8-10, 2014. Proceedings*, volume 8875 of *Lecture Notes in Computer Science*, pages 57–68. Springer, 2014.
8. J. F. Horty. *Agency and Deontic Logic*. Oxford University Press, Oxford, 2001.
9. S. Klarman and V. Gutiérrez-Basulto. Description logics of context. *Journal of Logic and Computation*, 2013.
10. Y. Murakami. Utilitarian deontic logic. In *Proceedings of the Fifth International Conference on Advances in Modal Logic (AiML 2004)*, pages 288–302, 2004.
11. K. Schild. A correspondence theory for terminological logics: Preliminary report. In *In Proc. of IJCAI-91*, pages 466–471, 1991.
12. R. A. Schmidt and U. Hustadt. First-order resolution methods for modal logics. In A. Voronkov and C. Weidenbach, editors, *Programming Logics – Essays in Memory of Harald Ganzinger*, volume 7797 of *LNCS*, pages 345–391. Springer, 2013.
13. E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, June 2007.
14. C. Wernhard and B. Pelzer. System description: E-KRHyper. In F. Pfennig, editor, *CADE-21*, volume 4603 of *LNAI*, pages 503–513. Springer, 2007.