ICRES 2018: International Conference on Robot
Ethics and Standards, Troy, NY, 20-21 August 2018.
https://doi.org/10.13180/icres.2018.20-21.08.001

PROBING FORMAL/INFORMAL MISALIGNMENT WITH THE LOOPHOLE TASK*

JOHN LICATO and ZAID MARJI

Advancing Machine and Human Reasoning (AMHR) Lab
Department of Computer Science and Engineering
University of South Florida
Tampa, FL, USA

Any autonomous agent deployed with some representation of rules to follow will face scenarios where the applicability of its given rules are not clear. In such scenarios, a malicious agent might successfully argue that some action which clearly goes against the spirit of the rules is allowed, under a strict interpretation of the rules. We argue that the task of finding such actions, which we call the *loophole task*, must be solved to some degree by an autonomous ethical agent, and thus is important for robot ethical standards. Currently, no artificially intelligent system comes close to solving the loophole task. We define this task, by characterizing it as exploiting a misalignment between informal and formal representational systems, and discuss our preliminary work towards creating an automated reasoner capable of solving it.

Keywords: Representations; Loopholes; Informal; Formal; Ethics

1. Introduction: Why Loopholes Matter to Ethics

Autonomous moral agents are typically deployed with some formal representation of the obligations constraining their allowed actions. These representations might be formulae in some highly formal language expressing obligations,[1–3] statutes of local, national, or international law written in legalistic language with varying levels of formality,[4–7] or even highly informal dictates expressed in natural language (e.g., "be good to humans"). It is difficult to imagine that any representational system, no matter how well-defined, can ever completely avoid the use of informal concepts (and complete rigidity of such rules, especially in the moral domain, may not be preferable anyway[8,9]). The problem is, these informal concepts introduce the possibility of loopholes—arguments that exploit the impreciseness of informal concepts in order to make the case that some formalization classifies some case in a way that goes against the intention of the formalization.

For example, Minnesota's 2007 "Freedom to Breathe Act" amended existing statutes so that tobacco products could no longer be smoked in public places. But an exception remained for "smoking by actors and actresses as part of a theatrical performance conducted in compliance with section 366.01."[a] The referenced section, however, did not define 'actor', nor 'theatrical performance.' Unsurprisingly, bars around the state soon organized "theater nights," in which customers were invited to attend and smoke, participating in imaginative, sometimes avant-garde performance pieces whose details varied from bar to bar.

These creative maneuvers did not stand up to court challenges.[b] Nevertheless, Minnesota's incompletely formalized statutes somehow opened themselves up to such loopholes,

*This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-16-1-0308. Any opinions, finding, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

[a] Ch. 144, Sec. 4167, Subd. 9, `https://www.revisor.mn.gov/statutes/?id=144.4167`

[b] `https://www.twincities.com/2009/07/13/appeals-court-bars-theater-nights-violated-smoking-ban/`

and it is important to understand why—especially for applications where such flexible interpretations can have serious consequences, as with autonomous moral robots. Perhaps most relevant to the ICRES community: We will, at some point, need to give autonomous robots a set of instructions formalizing allowed actions, whether in the form of laws, codes of ethics, or contracts encoded as machine code.[10] Is it then the case that no matter what, any formalization of obligations will lead to scenarios where the rules given are subject to loopholes that can be exploited?

We argue that the prognosis for formal representational systems is not hopeless. Although loopholes may be possible in every possible formalization, the lesson for AI researchers and policymakers is that more effort needs to be placed into (1) giving our autonomous reasoners the ability to reason about the rules they are given, and (2) finding ways to anticipate and close loopholes. Our lab is working on (1) through our project on *active formalization*,[11] but in this paper we will restrict our focus to our efforts addressing (2). As we will describe in Section 3, we believe it is possible to develop automated reasoning tools to assist in closing loopholes for any given formalization, by developing artificial reasoning systems capable of carrying out the *loophole task*, which we can attempt to define as follows:

Definition 1.1 (The Loophole Task). *Given a formal ruleset* $\mathbf{F} \in \mathscr{R}_F^*$*, meant to ban informal concept* $I \in \mathscr{R}_I^*$*, the Loophole Task is to find: (i) a case* C *that satisfies* I*, and (ii) informal arguments that* C *does not satisfy* $\mathbf{F}$*.*

Here, a *ruleset* refers to any partially formal classification mechanism. In the Minnesota smoking ban example, the stated purpose of the statute was to "protect employees and the general public from the hazards of secondhand smoke"[c], a phrase which uses the informal and difficult-to-define concepts 'protect' and 'hazards.' We might therefore describe the "theater nights" loophole as exploiting the misalignment between the formal ruleset (as defined by the statute) and the informal concept that the ruleset was designed to describe and ban.

2. Formalizing Loopholes

In this section, we will show how the loophole task can be thought of in terms of mixed-formality representational systems, thus leading to a more precise way of thinking about the Loophole Task. Definition 1.1 places the ruleset $\mathbf{F}$ and informal concept I as members of representational spaces $\mathscr{R}_F^*$ and $\mathscr{R}_I^*$, respectively. Our approach to the Loophole Task is to characterize it as exploiting a misalignment between elements of representational systems that are at different levels of formality—where $\mathscr{R}_F$ is a representational system which is more formal than $\mathscr{R}_I$. The terminology we use for representational objects comes from Ref. 11, which distinguishes between representational systems, representational spaces, and representations. Specifically:

Definition 2.1 (Representational System (RS)). *A representational system* $\mathscr{R}$ *is a tuple* $(\mathbf{M}, \mathbf{A})$*, where:*

- $\mathbf{M}$ - *A finite set of typed elements, called the members. Each member consists of a type and, optionally, a value. Types can either be primitive types (such as integer, boolean, string, etc.) or another representational system.*
- $\mathbf{A}$ - *A finite set of methods. Each method consists of a unique symbol and a method definition. If the method definition is empty, then the method is called an atomic method of the class.*

[c] `https://www.revisor.mn.gov/statutes/?id=144.412`

Definition 2.2 (Representation). *A representation $\mathcal{R}$ is a tuple $(\mathcal{R}_{inst}, sem)$, where:*

- *$\mathcal{R}_{inst}$ is an instantiated RS, which is an RS where all members are assigned values.*
- *sem is a "semiotic function" mapping the members and methods of $\mathcal{R}$ to the things they represent.*

An RS does not by itself represent, it only defines a space of possible representations. This allows us to separate the thing used to do the representing from the thing actually doing the representing, by making an analogy to object-oriented programming. Roughly: A class definition is to an object as a representational system is to a representation. For this reason, the above definitions are referred to as the "OO-inspired framework".[11]

For some representational system $\mathscr{R}$, the set of all possible representations it can produce (all possible ways to instantiate the class × all possible semiotic functions) is written $\mathscr{R}^*$, called $\mathscr{R}$'s "representational space". For convenience we write $\mathcal{R} \in \mathscr{R}^*$ when a representation $\mathcal{R}$ consists of an $\mathcal{R}_{inst}$ and *sem* in the space defined by RS $\mathscr{R}$.

2.1. *Interpreting Methods*

The OO-inspired framework allows us to clearly distinguish between many concepts that are often conflated in AI and AI-related fields: representations vs. representational systems vs. representational spaces, things members of a class can do vs. their properties, and so on. We can also compare representations at different levels of formality—but because it is outside the scope of this paper to mount a full defense of our view of formality,[d] it will suffice for now to define it as a partial ordering between representational systems, where $\mathscr{R}_F \geq_{LoF} \mathscr{R}_I$ if RS $\mathscr{R}_F$ is more formal than RS $\mathscr{R}_I$. We then introduce the following:

Definition 2.3 (Interpreting Method). *An interpreting method, in representation $\mathcal{R} \in \mathscr{R}^*$, is a method which (1) takes some description of a case C and evidence that C is an instance of symbol $\mathbf{s}$; (2) returns some measure of confidence that C is an instance of $\mathbf{s}$; and (3) is meant to serve as a way to recognize instances of symbol $\mathbf{s}$, as specified by $\mathcal{R}$'s semiotic function.*

Note that interpreting methods are not necessarily referentially transparent, particularly in informal RSes. For example, we might represent an individual human being as having some idea of how to recognize cats, but the algorithm-level description of how his inner mind works to determine whether or not a cat is present may not be available to him. Furthermore, note that the format of the case C and the evidence for C is specified by the RS to which the interpreting method belongs: a highly formal RS might require well-formed proofs as evidence, whereas a more informal RS might accept some combination of non-deductive arguments—these are called *interpretive arguments*, and come in a variety of forms, many of which have been catalogued by Refs. 12–14.

When an interpreting method always returns either 'True' or 'False,' we call it a *boolean interpreting method.* We can also say that a representation *recognizes symbol* $\mathbf{s}$ *through IM* if it has a boolean interpreting method *IM* meant to recognize $\mathbf{s}$. Finally, with all of these definitions in place, we can precisely state what we mean when we say that reasoners capable of solving the loophole task exploit the misalignment between RSes of differing levels of formality. First, observe that because of the way we have defined interpreting methods, it is entirely possible that two interpreting methods from different representations may recognize the same symbol, but fail to produce the same outputs on all possible inputs.

[d] We suspect that most commonly accepted senses of what it means for one representational system to be more formal than another can be expressed using the OO-inspired framework; proving this is a current project of our lab.

Now imagine that you are a lawmaker, hoping to ban some activity of which you only have an informal conceptual understanding. Your goal is to formalize this activity, in order to describe it in law. More precisely, let us assume that (1) the formal representation $\mathcal{F} \in \mathscr{R}_F^*$ is supposed to capture an informal representation $\mathcal{I} \in \mathscr{R}_I^*$ (i.e., the thing you want to ban), (2) $\mathscr{R}_F^* \geq_{LoF} \mathscr{R}_I^*$, and (3) both $\mathcal{F}$ and $\mathcal{I}$ recognize symbol **s** through boolean interpreting methods $\mathcal{F}.S$ and $\mathcal{I}.S$, respectively. Then, we can more precisely define the Loophole Task as finding cases C where:

Definition 2.4 (Overshooting/Undershooting). *$\mathcal{F}.S$ **overshoots** $\mathcal{I}.S$ on C when $\mathcal{F}.S$ returns True for case C, but $\mathcal{I}.S$ returns False. $\mathcal{F}.S$ **undershoots** $\mathcal{I}.S$ on C when $\mathcal{F}.S$ returns False for case C, but $\mathcal{I}.S$ returns True.*

3. A System for Finding Loopholes

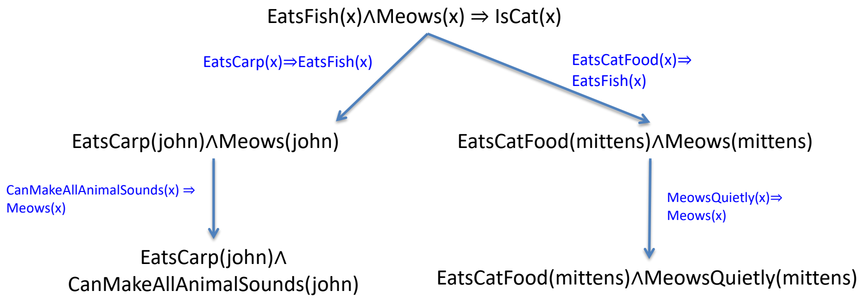
EatsFish(x)∧Meows(x) ⇒ IsCat(x)
EatsCarp(x)⇒EatsFish(x)
EatsCatFood(x)⇒ EatsFish(x)
EatsCarp(john)∧Meows(john)
EatsCatFood(mittens)∧Meows(mittens)
CanMakeAllAnimalSounds(x) ⇒ Meows(x)
MeowsQuietly(x)⇒ Meows(x)
EatsCarp(john)∧ CanMakeAllAnimalSounds(john)
EatsCatFood(mittens)∧MeowsQuietly(mittens)

Fig. 1. A simple warrant-reduction graph, which reduces a warrant (top) to specific cases (bottom) using reduction operators (in blue)

The rich definitions laid out in the previous section reveal how many pieces must fit together for a loophole to be found. A reasoner must essentially be able to produce a case, along with evidence across what might be RSes of completely different levels of formality. In the "Freedom to Breathe Act" example that opened this paper, the creative entrepreneur who first devised 'theater nights' must have been able to reason that theater nights would not fall under the legal definition using plausible legal reasoning, and simultaneously that theater nights does fall under the informal definition of "smoking-allowed nights that customers would want to attend". This ability to reason on two different levels simultaneously is far beyond the ability of any current AI—arguably, reasoning at even one of those two levels is already past the state-of-the-art.

All of this strongly suggests that solving and understanding the Loophole Task is not only worthwhile to the future of legal and ethical reasoning, but a highly non-trivial goal for artificial reasoning, and AI in general. The idea is that solving the Loophole Task can result in a tool to aid a formalization designer (e.g. a legislator, or a policy writer for autonomous moral agents, or a creator of a smart contract) by identifying possible loopholes that should be addressed before the formalization is deployed. Accordingly, our AMHR (Advancing Machine and Human Reasoning) lab at the University of South Florida has begun work on a system we believe will be able to make a dent in the problem, and the remainder of this paper will describe this work. However, we must temper expectations: at the time of this writing, this work is very preliminary.

Some loopholes can be found by exploiting the nature of *open-textured concepts*[15]—concepts whose extension is either underspecified, or are "highly dependent on context and human intentions".[16] There has been a wealth of work on solving the problem of open-textured concepts by combining rule-based and case-based reasoning.[16–19] But these, insofar as they can be classified as arguments from analogy or precedent, are only one type of interpretive argument (i.e., arguments that something should be interpreted a certain way[12]).

Our lab's approach draws from a modernization we are building of the warrant-reduction graphs (WRGs) described by Branting,[16] in order to automatically construct interpretive arguments that can be considered plausible loopholes to some formalization. Each WRG is essentially a large interpretive argument, consisting of many smaller interpretive arguments. If those interpretive arguments can be carefully selected according to the modes of evidence accepted by some interpreting method, then we essentially have a general-purpose tool to tackle the Loophole Task using the insights described in Section 2.

A WRG works by starting with a warrant, of the form $(c_1 \wedge ... \wedge c_n) \rightarrow P$. A case is a conjunction of facts $f_1 \wedge ... \wedge f_m$. The warrant graph determines whether the case is applicable to the warrant (and thus can be assigned the label P) by the use of *reduction operators*. For example, assume we are given the warrant "cats eat fish and meow," and the agent named 'Mittens' who has two features: He meows quietly, and eats cat food. One might be able to determine that the warrant applies to Mittens with the reduction operators "cat food contains fish" and "meowing quietly is meowing." A completed WRG constitutes a type of hybrid interpretive argument, consisting of multiple smaller interpretive arguments (depending on the sources of the reduction operators). This is illustrated in Figure 1, where the WRG is pictured as a tree, and each path from the root node to a leaf is an interpretive argument. However, Figure 1 also shows that this warrant will also allow one to argue that a human being who eats carp and can make animal sounds is also a cat.

We intend to explore answers to the following question: How far can we push the capabilities of WRGs as interpretive argument generators, drawing from partially structured datasets of formal and informal knowledge? Branting's WRGs relied on a manually collected corpus consisting of three types of data: warrants, cases, and reduction operators. Although his work achieved impressive results,[16] its use of small datasets limit its applicability (and its ability to generate interpretive arguments for the loophole task). Our proposed system to modernize WRGs in order to solve the Loophole Task is diagrammed in Figure 2. This project involves drawing from multiple semantic web databases[20–24] and recent advances in NLP and information extraction.[25–28] Both of these fields have seen major advances in the almost-20 years since Branting's publication.

Clearly there is much to be done. We also plan to generalize the way warrants and reduction operators are used in WRGs. As it stands they are currently horn clauses that do not allow negations, weighting individual conditions, modal operators, and so on.

References

1. L. Goble, *Logique et Analyse* **46**, 183 (2003).
2. P. McNamara, Deontic Logic, in *The Stanford Encyclopedia of Philosophy*, ed. E. N. Zalta (Stanford University, 2014) Winter 2014 edn.
3. S. Bringsjord, N. S. Govindarajulu, S. Ellis, E. McCarty and J. Licato, *Cognitive Systems Research* **28**, 20 (2014).
4. L. M. Friedman, *American Law: An Introduction*, 2 edn. (W.W. Norton and Company, Inc., 1998).
5. C. L. Cates and W. V. McIntosh, *Law and the Web of Society* (Georgetown University Press, 2001).
6. M. A. Pollack and G. Shaffer, The interaction of formal and informal lawmaking, in *Informal International Lawmaking*, eds. J. Pauwelyn, R. Wessel and J. Wouters (Oxford University Press, 2012)
7. H. Prakken, *Artificial Intelligence and Law* **25**, 341(Sep 2017).

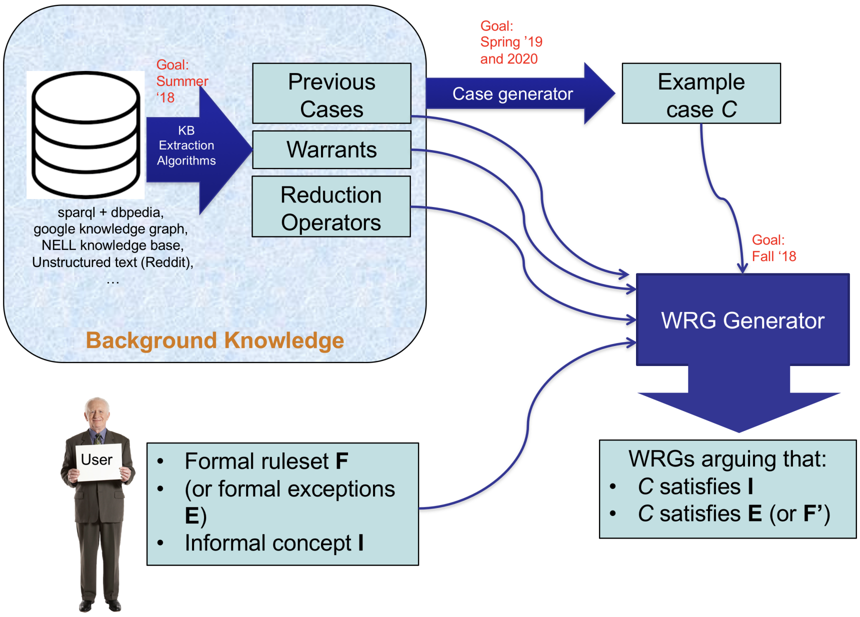

Fig. 2. Current plan for a proposed system to solve the Loophole Task

8. M. Guarini, *IEEE Intelligent Systems* **21**, 22 (2006).
9. M. Anderson and S. L. Anderson, *AI Magazine* **28**, 15 (2007).
10. N. Szabo, *Nick Szabo's Papers and Concise Tutorials* **6** (1997).
11. J. Licato and Z. Zhang, *Artificial Intelligence Review* **Forthcoming** (2018).
12. D. N. MarCormick and R. S. Summers, *Interpreting Statutes: A Comparative Study* (Routledge, 1991).
13. D. H. Berman and C. D. Hafner, Representing teleological structure in case-based legal reasoning: The missing link, in *Proceedings of the 4th International Conference on Artificial Intelligence and Law*, ICAIL '93 (ACM, New York, NY, USA, 1993).
14. G. Sartor, D. Walton, F. Macagno and A. Rotolo, Argumentation schemes for statutory interpretation: A logical analysis, in *Legal Knowledge and Information Systems. (Proceedings of JURIX 14)*, 2014.
15. H. Hart, *The Concept of Law* (Clarendon Press, 1961).
16. L. Branting, *Reasoning with Rules and Precedents: A Computational Model of Legal Analysis* (Springer, 2000).
17. K. D. Ashley and E. L. Rissland, *IEEE Expert* (Fall 1988).
18. K. E. Sanders, Representing and reasoning about open-textured predicates, in *Proceedings of the 3rd International Conference on AI and Law (ICAIL '91)*, 1991.
19. K. Forbus, T. Mostek and R. Ferguson, An Analogy Ontology for Integrating Analogical Processing and First-Principles Reasoning2002.
20. C. Matuszek, J. Cabral, M. Witbrock and J. DeOliveira, An introduction to the syntax and content of Cyc, in *Proceedings of the 2006 AAAI sprint symposium on formalizing and compiling background knowledge and its applications to knowledge representation and question answering*, 2006.
21. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. Ives, DBpedia: a nucleus for a web of open data, in *Proceedings of the 6th International Semantic Web Conference (ISWC2007)*, 2007.
22. K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in *Proceedings of the 2008 ACM SIGMOD International conference on Management of data (SIGMOD '08)*, (ACM, 2008).
23. D. DiFranzo, A. Graves, J. S. Erickson, L. Ding, J. Michaelis, T. Lebo, E. Patton, G. T. Williams, X. Li and J. G. Zheng, *Linking Government Data* **3**, 205 (2011).
24. C. Liang and K. D. Forbus, Learning Plausible Inferences from Semantic Web Knowledge by Combining Analogical Generalization with Structured Logistic Regression, in *Proceedings of*

the 29th AAAI Conference on Artificial Intelligence, 2015.

25. R. Socher, J. Bauer, C. D. Manning and A. Y. Ng, Parsing With Compositional Vector Grammars, in *Proceedings of ACL 2013*, 2013.
26. T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kociský and P. Blunsom, *CoRR* **abs/1509.06664** (2015).
27. M. Lippi and P. Torroni, *ACM Transactions on Internet Technology* **16** (2016).
28. A. Lai and J. Hockenmaier, Learning to predict denotational probabilities for modeling entailment, in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.