A Deontic Logic for Programming Rightful Machines: Kant's Normative Demand for Consistency in the Law

Ava Thomas Wright [1]

California Polytechnic State University San Luis Obispo

Abstract

In this paper, I set out some basic elements of a deontic logic with an implementation appropriate for handling conflicting legal obligations for purposes of programming autonomous machine agents. Relying on Immanuel Kant's philosophy of law, I argue that a deontic logic of the law should not try to work around such conflicts but, instead, identify and expose them so that the rights and duties that generate inconsistencies can be explicitly qualified and the conflicts resolved. Kantian justice demands that enforceable laws be consistent, precise, and minimally justifiable in a system. I then argue that a credulous, non-monotonic deontic logic can handle legal conflicts to satisfy these normative demands, with appropriate modifications. Finally, I propose an implementation of this logic via a modified form of "answer set programming," which I demonstrate with some simple examples. This proposed implementation helps advance the design of "rightful machines," autonomous machine agents that respect the authority of legitimate law.

Keywords: Conflicts, Deontic Logic, Non-Monotonic Logic, Kant, Law, Answer Set Programming, Logic Programming, Rightful Machines, Consistency, Standard Deontic Logic,

1 Rightful Machines

According to Immanuel Kant, appeals to reason alone cannot completely specify what our rights and duties with respect to each other are in disputed cases [7, 6:312]. In a society of moral equals, each person "has [her] own right to do what seems right and good to [her] and not to be dependent on another's opinion about this," Kant says [7, 6:312]. Hence even if everyone strives to act perfectly ethically with respect to others, rightful relations are impossible in the absence of a legitimate public authority, since "when rights are in dispute (ius controversum), there would be no judge competent to render a verdict having rightful force" [7, 6:312].

What is required, Kant argues, is

[1] avwright@calpoly.edu

> *a system of laws for a people...which because they affect one another, need a rightful condition under a will uniting them, a constitution (constituto)*, so that they may enjoy what is laid down as right. [7, 6:311]

Kant refers to this system of public laws and institutions as "public right" and a society existing under such a system as one existing in a "rightful" or "civil" condition, as opposed to a "state of nature." Only by constituting a *united will* to authoritatively define, enforce, and determine our rights and duties with respect to each other can we avoid injustice in inevitable cases of conflict between our rights, Kant argues (see [7, 6:313-14]). The determinations of a legitimate public authority as to the rights and duties of everyone interacting in community therefore generally take moral priority over individual ethical judgments in cases of conflict. To reject the authority of legitimate public law and institutions and instead use one's own private ethical judgment in such cases is to act wrongly, indeed, to commit wrong "in the highest degree," according to Kant [7, 6:308n].

Kant's philosophy of law is thus neither strictly positivist nor natural-legal. On the one hand, Kant argues that there is a necessary connection between positive law and morality because positive laws that violate conditions necessary to constitute the united will are illegitimate and, therefore, lack moral authority. [2] These conditions include, among others, respect for fundamental natural rights of freedom, independence, and equality as well as requirements that the laws be minimally rational and justifiable [10, p. 170-185]. A positive law that violates such fundamental rights or principles imposes no moral duty to obey it. But on the other hand, Kant argues that positive laws that do not violate such rights or principles are in general morally authoritative, even if those laws may otherwise be unwise or unjust. The positive laws of a legitimate public authority are necessary to cure the problem of indeterminacy of rights in the state of nature that makes rightful relations impossible. A positive law that is unjust may still impose a moral duty to obey it.

Kant's hybrid philosophy of law demands that enforceable laws be made consistent, precise, and minimally rationally justifiable in a system. These normative demands will shape and constrain how a deontic logic of the law should handle apparent conflicts between legal obligations. My aim in what follows is to set out the basic elements of such a deontic logic, together with an implementation that could reasonably be part of the programming of an autonomous machine agent. I first argue that a credulous, non-monotonic deontic logic can adequately satisfy Kant's normative demand for consistency in the laws, with appropriate modifications. I then propose an implementation of this logic via a modified form of "answer set programming," which I demonstrate with some simple examples. I hope this proposal will help advance the design of "rightful

[2] The authority of a law refers to whether citizens have a moral duty to obey it, whereas its legitimacy refers to the moral permissibility of enforcement of the law, regardless of whether citizens have a duty to obey it or not. Kant's position is that the law's legitimacy is a necessary condition for its authority.

machines," autonomous machine agents that properly respect the authority of the law within the bounds of fundamental rights and principles of justice [12] [13].

2 Conflicts Between Legal Obligations

2.1 The Normative Demand for Consistency in the System of Laws

Kant appears to deny that there can ever be conflicts between legal obligations:

> [S]ince duty and obligation are concepts that express the objective practical necessity of certain actions, and two rules opposed to each other cannot both be necessary at the same time—rather if it is one's duty to act according to one of them, to act according to the opposite one is not only no duty, but even contrary to duty—a collision of duties and obligations is not even conceivable (obligationes non colliduntur). [7, 7:224]

Kant argues here that if one were required to perform an action in accordance with an obligation (Oa) that opposed another simultaneous obligation prohibiting the action ($O \sim a$), then acting in accordance with the first obligation (a) would imply acting in a way that violated the second obligation ($\sim a$), a performance that is not even conceivable ($a \wedge \sim a$). One cannot be obligated to perform what is impossible ($O(a \wedge \sim a)$); therefore, Kant concludes, one cannot simultaneously be subject to opposing obligations ($Oa \wedge O \sim a$). (Here "O" is a monadic operator for an obligation one has; "a" is an action one performs.)

Kant's claim that obligations cannot come into conflict ($\sim (Oa \wedge O \sim a)$) may be understood either descriptively or normatively. Understood descriptively, the claim seems false. There seems to me no reason to think that even a thoroughly rational public authority might not create legal obligations that contradict in situations that authority did not foresee. For example, suppose a state authority enacts a traffic law that requires stopping at stop signs and also another law that forbids stopping in front of military bases (see [9] : 179). It is not inconceivable that a local government agency might then erect a stop sign in front of a military base, creating a conflict of narrow legal obligations under applicable enforceable laws for drivers unfortunate enough to encounter the situation. The possibility of such conflicts seems a mundane descriptive fact about any system of laws, and while one might be tempted to assert that the ordinances in question cannot be held to conflict in the case because the driver can have only one true legal obligation, this assertion seems clearly normative rather than descriptive.

Kant's claim that legal duties cannot conflict should be understood as a normative constraint on the prescriptive system of enforceable legal obligations. At the *descriptive* level, law contains contradictory obligations; but at the *prescriptive* level of enforceable obligations, the laws should be completely consistent. What does Kant's normative demand for consistency in the system of enforceable public laws imply for a deontic logic of the law?

2.2 The Inadequacy of Standard Approaches to Legal Conflict

The standard system of deontic logic (SDL) is a normal modal logic with a deontic gloss on the $\Box$ (box) and $\Diamond$ (diamond) operators, interpreted as obligation (O) and permission (P), respectively. The system is a K logic characterized, syntactically, by the D (deontic) axiom, '$\Box p \rightarrow \Diamond p$' (that is, if action p is obligatory, then p is permitted, $Op \rightarrow Pp$) or the 'D$\Diamond$Introduction' rule in a Fitch-style proof system, and, semantically, by a seriality condition on frames in the Kripkean possible world semantics (that is, for every world, there is at least one accessible world). What SDL amounts to is the rejection of conflicts of obligation ($\sim (\Box p \wedge \Box \sim p)$), which is just the D axiom.

But since, as we have seen, there is no reason to think that deontic conflicts do not occur in the law as a descriptive matter, an adequate deontic logic of the law should not deny the possibility of such conflicts, as SDL does. Yet if one rejects axiom D so as to admit conflicts of obligation into SDL, then the logic becomes immediately incoherent, since given some standard principles for the inheritance of obligations (RM) (If $\vdash p \rightarrow q$, then $\vdash Op \rightarrow Oq$) and aggregation (AND) ($\vdash (Op \wedge Oq) \rightarrow O(p \wedge q)$), one can derive any obligation from the contradiction in accordance with the classical logical principle ex falso quodlibet (EFQ) ($(p \wedge \sim p) \rightarrow q$) [5, p. 463–4]. That is, given a dilemma where simultaneously Op and $O \sim p$, any arbitrary action q can be proven to be obligatory. (For example: 1. Op. assp. 2. $O \sim p$. assp. 3. $O(p \wedge \sim p)$. 1,2 AND. 4. $(p \wedge \sim p) \rightarrow q$. EFQ. 5. $O(p \wedge \sim p) \rightarrow Oq$. 4 RM. 5. Oq. 3,4 MP.) A number of efforts to weaken one or more of these principles in order to avoid the deontic explosion of arbitrary obligations have therefore been undertaken, though with limited success.

Semi-classical and paraconsistent logics avoid this inferential explosion by replacing the two truth values (true, false) of classical semantics with a semantics of many values (e.g., null, just true, just false, and both true and false) [4, p. 99–105, 195–196]. These logics have been thought too weak to be very useful, however, because they fail to vindicate certain common, intuitively valid deontic arguments. For example: 1. S ought to fight in the war or perform alternative service to his country ($O(f \vee a)$). 2. S ought not fight ($O \sim f$). 3. Therefore, Smith ought to perform alternative service to his country (Oa) [5, p. 467]. This intuitively valid conclusion cannot be derived in most paraconsistent or relevance deontic logic systems because they lack the disjunctive syllogism of propositional calculus needed to make the inference $(f \vee a) \wedge \sim f \rightarrow a$. Such failures are not conclusive, however, and overcoming them continues to be an area of active research.

Other efforts to describe contradictions while avoiding deontic inferential explosion attempt to do so by weakening Aggregation (AND) or Inheritance of Obligations (RM), rather than by rejecting classical EFQ. They typically do so by imposing prior consistency or permissibility checks. For example, Aggregation (AND) may be weakened by requiring that p and q be jointly possible before allowing their aggregation under obligation (CAND: If $\nvdash p \rightarrow \sim q$ then $\vdash (Op \wedge Oq) \rightarrow O(p \wedge q)$), or by requiring that p and q be jointly permissible (PAND: $\vdash P(p \wedge q) \rightarrow ((Op \wedge Oq) \rightarrow O(p \wedge q))$). Inheritance of Obligations (RM) may be weakened by requiring that p be permissible before allowing q to inherit an obligation from the obligation that p (RPM: If $\vdash p \rightarrow q$ then $\vdash Pp \rightarrow (Op \rightarrow Oq)$) [5, p. 467–473]. Each resulting logic avoids deontic inferential explosion and has its relative advantages and disadvantages in accounting for the intuitive validity of various example deontic arguments.

2.3 What Kant's Normative Demand for Consistency in the Laws Really Requires

The problem with both paraconsistent logics and these other efforts to weaken SDL's axioms for purposes of a deontic logic of the law, however, is that they offend the demand for consistency understood as a *normative* requirement. Contradictory prescriptive legal obligations are admitted as first-class citizens of such logics. In paraconsistent logics, inferences are derived in the face of such contradictions by the alchemy of a non-classical semantics, which often confounds intuitions. In the weakened deontic logics described above, by contrast, contradictions are like icebergs around which reasoning proceeds gingerly, if at all. In neither case does the logic require that one contradictory obligation be defeated, or that rules generating the contradiction be qualified or revised, in order to allow an inference through the other obligation, or vice versa.

For example, suppose that a criminal statute requires the punishment of anyone who intentionally kills a person ($k \rightarrow Op$), while another statute forbids punishing minors ($m \rightarrow O \sim p$), and suppose a court confronts a case where a minor has intentionally killed someone ($k \wedge m$) (see Alchourron 1991). This licenses the inferences Op and also $O \sim p$, so creating a conflict of obligations. The weakened logics above draw both inferences but then limit further inferences that depend directly on one or another of them. For example, suppose that punishment always consists in incarceration ($p \rightarrow c$). RM would license $Op \rightarrow Oc$, and therefore the inference that the killer ought to be incarcerated, despite that she is a minor (Oc) and ought not to be punished ($O \sim p$). The weakened RPM logic above appropriately blocks this inference because Op is impermissible, $O \sim p$ (where $O \sim p \leftrightarrow \sim Pp$). The RPM logic infers that there is a killer who is a minor (k, m), and that the court is obligated to punish her (Op) and obligated not to punish her ($O \sim p$), but then blocks the explosion of further inferences such as that she ought to be incarcerated (Oc). While the RPM logic thus succeeds in admitting conflicts while avoiding a deontic explosion of inferences, which is its goal, its approach to doing so seems to me to miss the point of admitting deontic conflicts in the first place.

Conflicts between deontic obligations should stimulate rational inference rather than shut it down. What conflicts normatively indicate in a legal deontic context is that one must either revise one or the other of the inconsistent formulas, or prioritize one over the other or, semantically, that one must choose between competing consistent models of (revised) rules given the facts of some conflict situation. While a doxastic or epistemic application of modal logic may perhaps not be subject to the same normative demands, a deontic logic of the law must provide some mechanism to adjudicate between consistent sets of formulas. The goal in the case of the killer who is a minor above should be to render a judgment as to whether her punishment is consistent with everyone's obligations and rights in the system of public laws, subject to constitutional constraints. But paraconsistent logics and weakened deontic logics that admit contradictions seem useless for this purpose.

A court might resolve the case by, for example, qualifying the rule against homicide so as not to apply to minors $(k \wedge \sim m) \rightarrow Op$, or, on the other hand, by qualifying the rule barring the punishment of minors so as not to apply in cases of intentional homicide $(m \wedge \sim k) \rightarrow O \sim p$, or the court might articulate some explicit rule of priority [1, p. 423–424]. A deontic logic of the law should be able to admit the conflict descriptively and provisionally generate inferential alternatives, together with further consequences, in order to evaluate each resulting consistent set of rules and require a decision. The weakened deontic logics above instead simply admit the conflict and limit further inferences. What is needed is a deontic logic that admits the presence of contradictions at a descriptive level but whose semantics insists that they be authoritatively resolved at the prescriptive level of enforceable public laws. This resolution should render our legal obligations precise, and, moreover, must be minimally rationally justifiable by reference to authoritative laws, orders, or judgments.

In the next subsection, I argue that a non-monotonic reasoning system with a classical (rather than paraconsistent) base can meet these normative requirements, with appropriate modifications.

2.4 Non-Monotonic Deontic Logics and the "Credulous" Reasoning Semantics

Non-monotonic reasoning systems (NMRs) are able to admit contradictions without igniting a deontic inferential explosion of obligations because they reject monotonicity (i.e., "If $K' \vdash p$ and $K' \subseteq K$, then $K \vdash p$."). What the rejection of monotonicity means is that some inferences might no longer be drawn when new premises are introduced; for example, one might introduce a new fact that directly contradicts some fact upon which an inference depends, so defeating that inference. NMRs thus avoid the deontic inferential explosion of obligations that plagues SDL; at the same time, NMRs insist that the set of consequences inferred be consistent.

Classical logic can be defined as a structure $S = (F, R)$ where F is a set of formulas and R is a set of rules of inference. R defines a classical consequence relation ('$\vdash$') between a set of formulas and a formula of the language (p). A non-monotonic logic can be defined as a structure $S = \{F, K, R\}$ where F is a set of formulas, K is a set of default rules, and R is a set of rules of inference that define a non-monotonic consequence relation ('$|\!\sim$' note the "snake"). This consequence relation may be defined simply as follows [8, p. 66–67]:

$F, K| \sim p$ if and only if $F, K' \vdash p$ for all subsets $K' \subseteq K$ which are maximally consistent with F.

A subset K' of K is maximally consistent with F if and only if it is consistent with F and there is no superset with F that is consistent and also a subset of K.

For example, suppose $K = \{b \rightarrow f, p \rightarrow\sim f\}$ (“Birds fly; penguins do not fly.”). Suppose $F = \{b\}$. Hence $\{b\} \mid\!\sim_K f$ because for all subsets K' of K that are maximally consistent with $\{b\}$, $\{b\}, K' \vdash f$. That is, $\{b, b \rightarrow f, p \rightarrow\sim f\} \vdash f$. Suppose b: “chilly is a bird;” therefore, f: “chilly flies” because $b \rightarrow f$: “birds fly.” But now if we also add p (p: “chilly is a penguin”) to F, then $\{b, p\} \not\mid\!\sim_K f$ because $\{b, p\} \nvdash_K f$ for all subsets K' of K that are maximally consistent with $F = \{b, p\}$; that is, while $\{b, p, p \rightarrow f\} \vdash f$, the subset $\{b, p, p \rightarrow\sim f\} \nvdash f$. (Note that $\{b, p\} \not\mid\!\sim_K \sim f$, either, because $\{b, p, b \rightarrow f\} \nvdash \sim f$. We cautiously infer neither that chilly flies nor that chilly does not fly.) This demonstrates that adding p to the premises causes the conclusion f to be withdrawn in the face of contradiction.

What should the semantics of a deontic NMR appropriate for handling conflicts between enforceable legal obligations be? The consequences a NMR draws given some set of rules K and first-order formulas F can be defined in terms of the “extensions” of (K, F), which, informally, are the rational and justifiable sets of conclusions one can draw given (K, F). Extensions are rational in the sense that conclusions are not accepted if they would create inconsistencies, and justifiable in the sense that 1) all the conclusions that one accepts have some justification in (K, F), while 2) adding any further conclusion would create an inconsistency. “Credulous” reasoning defines the consequences as those in exactly one extension; “skeptical” reasoning defines the consequences as those that lie at the intersection of all extensions. “Ideally” skeptical reasoning defines consequences in terms of the intersection of paths of support, where facts and rules are understood to form an inheritance network [11].

I argue that a credulous rather than skeptical reasoning semantics is needed for a deontic logic of the law. Consider again the case of the killer who is a minor $(k \wedge m)$, where killers ought to be punished $(k \rightarrow \text{Op})$ and minors ought not to be $(m \rightarrow\sim \text{Op})$. The extensions of these rules and facts are $\{k, m, \text{Op}\}$ and $\{k, m, \sim \text{Op}\}$. Skeptical reasoning cautiously infers as consequences only that there is a killer who is a minor (the intersection of the extensions, $\{k, m\}$), but whether punishment is obligatory or not is left undefined. Yet Kantian justice requires an authoritative ruling in the case; otherwise, any enforcement (or lack of it) is wrongfully coercive. Obligations must be determinate at the prescriptive level of enforceable law—punishment either is or is not obligatory in the case.

Credulous reasoning appropriately requires *exactly one* or the other conclusion as a consequence, which will then persist in the knowledge base to guide and constrain further inferences. For example, suppose we again add $(p \rightarrow c)$ (“punishment is by incarceration”) to our knowledge base of rules. If the NMR credulously concludes that punishment is obligatory (Op) in the case—perhaps on the theory that minors should be punished for crimes that are felonies such as murder—then the obligation to incarcerate (Oc) will also follow as a consequence. If the NMR credulously concludes, on the other hand, that the minor should not be punished ($\sim$ Op)—even in felony cases—then Oc does not follow as a consequence. This correctly reflects whether and how further inferences such as Oc should be reached.

3 Proposal: Answer Set Programming Rightful Machines

In this section, I propose an approach to programming autonomous machine agents to handle conflicts between legal obligations via a form of logic programming referred to as "answer set programming," which can be viewed as a efficient machine implementation of nonmonotonic formalisms. Answer set programming, with some modifications, can therefore capture the deontic logic of the law I proposed in in the previous section.

The two dominant semantics for extended logic programs are the "answer set" / "stable model" semantics and the "well-founded model" semantics. The answer set semantics for logic programs defines a logic program's consequences in terms of the intersection of its answer sets (extensions), however, and the well-founded model will appear as a subset of this intersection. Both semantics thus reflect skeptical reasoning, where the well-founded semantics reflects "ideally" skeptical reasoning. To achieve credulous reasoning, I exploit only a part of the answer set semantics to 1) enumerate answer sets, and then 2) require the selection of exactly one answer set. This credulous semantics will achieve the main aim of admitting conflicts at the descriptive level while normatively requiring their resolution at the prescriptive level of enforceable law.

3.1 Background: Answer Set Semantics for Logic Programs

First, I briefly sketch the answer set semantics for extended logic programs. My aim here is to provide some background material on how answer set programming works, rather than a formal treatment (see [3] for a formal account). A logic program (Π) consists of a set of rules of this form:

$$\texttt{a :- b}_k\texttt{, b}_{k+1}\texttt{, ..., b}_m\texttt{, not c}_{m+1}\texttt{, ..., not c}_n\texttt{.}$$

where k, m and n are non-negative integers, a, b, and c are atomic formulas or their negations (i.e. p or $-p$), and "not" is "negation-by-failure." a above is referred to as the head of the rule, while formulas following the ":-" symbol make up the body of the rule. A rule with no head is a constraint, while a rule with no body is a fact.

The answer sets of a logic program Π are obtained by the following procedure [3]. I will first provide a general description of the procedure and then apply it to an illustrative example. First, generate a partial interpretation (I) of Π, which is a consistent set of ground literals formed from the rules of Π. Ground rules, literals, and terms of a logic program contain no variables; hence to create a ground instance of a rule of Π, replace all the rule's variables with ground terms of Π. A literal p is true if it is an element of the interpretation and false if its complement is; otherwise, the literal is undefined. Next, obtain the "reduct" of the program Π with respect to the generated partial interpretation I of Π. The reduct of Π is obtained by first deleting every rule from Π with "not p" in its body, where p is a member of I, and then deleting all "not q" from the remaining rules of Π, where q is any literal. Finally, repeatedly obtain the immediate consequences ($T(I)$) of the reduct by applying modus ponens and avoiding contradictions until reaching a fixpoint (Cn) where the set of immediate consequences no longer changes. If the set of immediate conclusions is the same as the partial interpretation I, then the set is an answer set. Repeat this procedure until you have found all answer sets.

3.2 Encoding Alchourron's Example of the Killer who is a Minor

To illustrate, suppose we want to apply an answer set programming approach to the conflict adduced previously between defeasible legal rules that 1) killers should be punished, but 2) minors should not be punished, and the case is such that a minor has killed someone [1]. How best to capture and resolve this case of conflict between legal obligations in answer set programming? We want to know what the rational, justifiable sets of conclusions (answer sets) are in the case, given the facts and applicable legal rules. Intuitively, there are two such sets: either 1) the minor should be punished because she is a killer, despite being a minor, or 2) the minor should not be punished because she is a minor, despite that she is a killer.

The following intuitive encoding of the rules and facts is inadequate because it has just one answer set consisting only in the initial facts ($\{k, m\}$):

```
k.  m.            % a killer who is a minor
p :- k, not m.    % punish killers unless they are minors
-p :- m, not k.   % do not punish minors unless they are killers
```

Encoding the laws in conflict as "normal" default rules, on the other hand, generates the results we want:

```
k.  m.            % a killer who is a minor
p :- k, not -p.   % punish killers unless they should not be
-p :- m, not p.   % do not punish minors unless they should be
```

This encoding yields two rational, justifiable extensions (answer sets). Either 1) the killer who is a minor should be punished ($\{k, m, p\}$), or 2) the killer who is a minor should not be punished ($\{k, m, -p\}$).

The following table displays how the procedure described above finds answer sets for this program (answer sets are starred):

Interpretation (I)	**Reduct (P^I)**	$\boldsymbol{\gamma p(I)}$
{}	k. m.	k, m
k	k. m. p :- k. -p :- m.	⊥
m	k. m. p :- k. -p :- m.	⊥
k, m	k. m. p :- k. -p :- m.	⊥
k, m, p	k. m. p :- k.	k, m, p*
k, m, -p	k. m. -p :- m.	k, m, -p*

3.3 Regimenting the Proposed Encoding

While encoding legal rules as normal default rules achieves correct results, the encoding should provide some way to make the qualifications on the rules that generated each answer set explicit. This will meet Kant's normative demand that enforceable legal obligations be made precise. The following intuitive encoding of rules with explicit qualifications, however, has no answer sets:

```
k.  m.              % a killer who is a minor
p :- k, not q1.   % punish killers unless this rule is qualified
-p :- m, not q2.  % do not punish minors unless this rule is
    qualified
```

I propose the following encoding, which explicitly tracks the qualifications required as they appear in each answer set and, moreover, leaves room to later make additional qualifications to the rules as needed:

```
k.  m.  a.     % a killer who is a minor (and an action is taken)

p :- k, not q1.    % rule 1: punish killers
q1 :- a, not p.    % unless this rule is qualified

-p :- m, not q2.   % rule 2: do not punish minors
q2 :- a, not -p.   % unless this rule is qualified
```

The following table indicates how answer sets for this program are found. (Partial interpretations are generated and tested here in order by set inclusion; '...' indicates interpretations omitted to save space.)

Interpretation (I)	**Reduct (P^I)**	$\boldsymbol{\gamma p(I)}$
{}	k. m. a. p :- k. q1 :- a. -p :- m. q2 :- a.	⊥
...	...	...
k, m, a, p	k. m. a. p :- k. -p :- m. q2 :- a.	⊥
...	...	...
k, m, a, q1, q2	k. m. a. q1:- a. q2 :- a.	k, m, a, q1, q2*
k, m, a, q1, -p	k. m. a. q1:- a. -p :- m.	k, m, a, q1, -p*
k, m, a, q2, p	k. m. a. p :- k. q2 :- a.	k, m, a, q2, p*
k, m, a, q1, q2, p	k. m. a. q2 :- a.	k, m, a, q2
...	...	...

Answer sets for this program are $\{k, m, a, q1, q2\}$, $\{k, m, a, q1, -p\}$, and $\{k, m, a, q2, p\}$. These answer sets reflect explicit qualifications made on both rules $(q1, q2)$, and then on one rule $(q1)$, or the other $(q2)$. In this encoding, qualifications on legal rules are triggered when there is an action (a) and the negation of the head of the rule qualified (p) is by default negation (e.g., `not p`), rather than classical negation $(-p)$, as in the first attempt at encoding qualifications above. This encoding thus achieves the same behavior as normal default rules but with explicit qualifications on rules. Each answer set reflects a rational, justifiable set of conclusions one might draw, given the facts and applicable rules in the case; moreover, the rule qualifications necessary to construct each answer set are now made explicit in the set.

While qualifying both rules yields a set of consequences that are both consistent and justifiable (i.e., the set stating merely that there is a killer who is a minor, $\{k, m, a, q1, q2\}$), doing so fails to meet the Kantian normative demand that legal rights in conflict cases must be resolved. The main purpose of the law on Kant's account is to rightfully resolve disputes over our rights and duties with respect to each other. To meet this normative demand, a "ruling" predicate is added to the encoding that requires each answer set to provide a definite determination of the rights and obligations in conflict:

```
% a ruling is required
:- not ruling.
ruling :- p.  ruling :- -p.
```

This eliminates the first answer set that qualifies both rules $\{k, m, a, q1, q2\}$, and thus fails to determine an answer to the question whether the killer who is a minor should be punished, or not. The credulous reasoning semantics will now demand that exactly one of the two remaining answer sets ($\{k, m, a, q1, -p\}$, $\{k, m, a, q2, p\}$) be selected as the program's consequences.

3.4 Simple Legal Case: A Shooting in Self-defense

I illustrate the proposed encoding by evaluating a conflict between the legal rule barring murder and one permitting the use of force in self-defense. Imagine a case of first impression that generates a conflict between these rules. In what follows I use the lparse grounder and clingo parser to ground and solve encoded logic programs (see [2] for details).

First, a situation is described with facts that may satisfy elements of various legal theories and rules:

```
%%% conflict situation
intentional(shooting).  act(shooting).  causes_death(shooting,
    someone).  person(someone).
attacked(me).  force(shooting, me).  retreated(me).
```

The facts as encoded in this example are necessarily somewhat stipulative. An actual autonomous machine agent governance system would include perceptual and other subsystems organized to generate new facts (beliefs), which might then generate messages for processing at progressively higher levels in the rational agent hierarchy. My aim here is to isolate and describe only the legal reasoning part of the system.

Applicable legal rules are then extracted and encoded, perhaps by interpreting a semantic legal knowledge base. The head of a legal rule is encoded as a deontic prescription on an action (e.g., that an action is obligatory ($ob(A)$), forbidden ($ob(-A)$), permissible ($pe(A)$), or omissible ($pe(-A)$)). The body of the rule will then invoke legal theories relevant to establishing the deontic status of the action (e.g., that the action constitutes a murder, or is self-defense, an action by omission, an act of necessity, etc.), where elements establishing these theories are define in existing statutory or case law (e.g., the common law rule that a murder is an intentional act that causes the death of a person).

Here is the first part of the body of a simplified rule prohibiting murder:

```
ob(-A) :- murder(A).
% legal elements of murder: malice killing
murder(A) :- intentional(A), act(A), causes_death(A, P), person(P).
```

The body of the rule is then completed with a generic defeasible qualification (e.g. $qual(r1(A))$) on the rule tagged with the rule number ($r1$):

```
ob(-A) :- murder(A), not qual(r1(A)).
qual(r1(A)) :- act(A), not ob(-A).
```

This preserves the possibility that the rule may be defeated for reasons other than those explicitly anticipated in the local context of the system.

Here accordingly are legal rules prohibiting murder ($r1$) and a conflicting rule permitting the use of force in self-defense (r2):

```
%%% r1: it is obligatory not to murder
ob(-A) :- murder(A), not qual(r1(A)).
qual(r1(A)) :- act(A), not ob(-A).

% legal elements of murder: malice killing
murder(A) :- intentional(A), act(A), causes_death(A, P), person(P).

%%% r2: it is permissible to use force in self-defense
pe(A) :- self_defense(A), not qual(r2(A)).
qual(r2(A)) :- act(A), not pe(A).

% legal elements of self-defense: use of force by one who is
    attacked and tried to retreat
self_defense(A) :- force(A, P), attacked(P), retreated(P).
```

The legal rules provided here are obviously simplified in order to isolate how the system handles conflict. For example, the legal intention required to establish the mens rea for murder is "malice aforethought," which implies at least a reckless awareness that one's act will cause the death of a person (see [6]). I omit such details here. [3]

Next, a constraint to avoid conflicts between deontic obligations within answer sets is added as well as standard deontic implication and convenient equivalence relations between obligations and permissions. These reflect axioms and equivalences of SDL that apply within answer sets.

```
% avoid deontic conflict
:- ob(A), ob(-A).

% deontic implication (D)
pe(A) :- ob(A).   % obligation implies permission

% inheritance of obligation (RM)
ob(A) :- ob(B), -A.
```

[3] I do not mean to imply that these details cannot be supplied, however—indeed, they must be, if the law is to meet the normative requirement that prescriptive obligations be precisely specified.

```
ob(A) :- ob(B), B.
A :- B.

% deontic equivalences
ob(A) :- -pe(-A).  -pe(-A) :- ob(A).
pe(A) :- -ob(-A).  -ob(-A) :- pe(A).
ob(-A) :- -pe(A).  -pe(A) :- ob(-A).
pe(-A) :- -ob(A).  -ob(A) :- pe(-A).
```

Finally, an updated "ruling" constraint requiring each answer set to provide some deontic resolution of the legal conflict at issue completes the program. Appropriate `#show` directives have also been added to avoid clutter.

```
% a ruling is required: an obligation, prohibition, permission, or
    omission
:- not ruling.
ruling :- ob(A).  ruling :- ob(-A).     % obligation, prohibition
ruling :- pe(A).  ruling :- pe(-A).     % permission, omission
#show pe/1.  #show ob/1.  #show qual/1.
```

The program generates the following answer sets. These answer sets represent the rational and justifiable sets of consequences with associated explicit rule qualifications that one might infer in the case, given the facts and applicable law as encoded.

```
Answer: 1
murder(shooting) self_defense(shooting) ob(-shooting) pe(-shooting)
    qual(r2(shooting))
```

```
Answer: 2
murder(shooting) self_defense(shooting) qual(r1(shooting))
    pe(shooting)
```

Either (Answer: 1) one is *obligated not* to intentionally shoot to kill, `ob(-shooting)`, in the case because the self-defense rule (r2) is qualified not to apply here, `qual(r2(shooting))`. Or (Answer: 2) one is *permitted* to intentionally shoot to kill, `pe(shooting)`, because the murder rule is qualified not to apply in the case, `qual(r1(shooting))`. The law as stipulated is thus conflicted with respect to whether the shooting is forbidden or permitted.

Now, it has long been established in the criminal law that a killing that would otherwise be a murder is justified if the killing meets the elements of self-defense. Hence the conflict here should be resolved by explicitly qualifying the murder rule (r1) to permit killing in self-defense (Answer: 2). To do that, the self-defense qualification is made explicit, while retaining the generic defeasible qualification on the rule so that it remains a candidate for defeat in future cases of conflict (e.g., with other defenses such as necessity or excuses such as insanity). The murder rule (r1) is thus adjusted as follows (changes in bold):

```
%%% r1: it is obligatory not to murder
ob(-A) :- murder(A), not qual(r1(A)), not qual(r21(A)).
qual(r1(A)) :- act(A), not ob(-A).

qual(r21(A)) :- self_defense(A).  % unless in self-defense
```

With this adjustment, the program produces only one answer set in the situation described, where the murder rule is qualified, `qual(r21(shooting))`, because the case as stipulated is clealry self-defense.

```
Answer: 1
murder(shooting) qual(r21(shooting)) self_defense(shooting)
    qual(r1(shooting)) pe(shooting)
```

The shooting in this case is therefore permissible. Further qualifications on the updated murder rule may be progressively entertained and accepted or rejected as new cases arise, such as qualifications for excuses or defenses such as insanity, necessity, etc.

The total number of possible answer sets in a case of conflict will be the power set of the available qualifications; in this simple example, $P(\text{qual}(r1()), \text{qual}(r2()))$, or only four total sets, including the empty set. The answer set semantics eliminates inconsistent combinations of rule qualifications and inferences that violate non-contradiction and modus ponens. The `ruling` predicate eliminates the empty set and the set where no decision is made because all rules are qualified. If there are multiple answer sets remaining, then the credulous reasoning semantics will require the selection of exactly one.

The resulting set of enforceable legal obligations will be a rational and justifiable set of applicable laws explicitly qualified to resolve inconsistencies and so to precisely determine legal obligations and rights in the case. So long as the laws in the set are also legitimate, an autonomous machine agent that acts in accordance with them is a rightful machine.

References

[1] Alchourron, C., *Conflicts of norms and the revision of normative systems*, Law and Philosophy **10** (1991), pp. 413–425.

[2] Gebser, M., R. Kaminski, B. Kaufmann and T. Schaub, "Answer Set Solving in Practice," Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2012, see https://potassco.org/.

[3] Gelfond, M., *Chapter 1: Answer sets*, Foundations of Artificial Intelligence **3** (2008), pp. 285–316.

[4] Girle, R., "Modal Logics and Philosophy, 2d ed." Montreal, MQUP, 2017.

[5] Goble, L., *A logic for deontic dilemmas*, Journal of Applied Logic **3** (2005), pp. 461–483.

[6] Institute, A. L., "Model penal code: official draft and explanatory notes: complete text of model penal code as adopted at the 1962 annual meeting of the American Law Institute at Washington, D.C., May 24, 1962," The Institute, Philadelphia, 1985.

[7] Kant, I., *The metaphysics of morals (1797)*, in: P. Guyer and A. Wood, editors, *The Cambridge Edition of the Works of Immanuel Kant*, Cambridge University Press, Cambridge, 1992 Translated by M. Gregor. Citations to Kant's work are given using standard Academy pagination.

[8] Maranhao, J., *Why was alchourron afraid of snakes?*, Analisis Filosofico **26** (2006), pp. 62–92.

[9] Navarro, P. and J. Rodriguez, "Deontic Logic and Legal Systems," Cambridge University Press, Cambridge, 2014.

[10] O'Neill, O., "Constructing Authorities," Cambridge University Press, Cambridge, 2011.

[11] Touretzky, D., J. Horty and R. Thomason, *A clash of intuitions: The current state of nonmonotonic multiple inheritance systems*, in: *Proc. IJCAI-87*, Morgan Kaufmann, 1987, pp. 476–482.

[12] Wright, A., *Rightful machines*, in: H. Kim and D. Schönecker, editors, *Kant and Artificial Intelligence*, Walter de Gruyter GmbH & Co KG, 2021 .

[13] Wright, A. T., *A kantian course correction for machine ethics*, in: G. J. Robson and J. Y. Tsou, editors, *Technology Ethics: A Philosophical Introduction and Readings*, Routledge, New York, 2023 pp. 141–151.