Deontic Ambiguities in Legal Reasoning

Guido Governatori
Brisbane, Queensland, Australia
guido@governatori.net

Antonino Rotolo
University of Bologna
Bologna, Italy
antonino.rotolo@unibo.it

ABSTRACT

What happens if the way in which we handle a genuine deontic conflict —i.e., a deontic ambiguity— matters regarding the application of other norms that are not directly affected by that conflict? We argue that the law requires sometimes propagating the ambiguity to other norms and sometimes confining it to some norms only. We explore this issue and model different reasoning patterns. The problem is addressed in a new variant of Defeasible Deontic Logic. The contribution of this paper is threefold: (a) we extend the treatment of ambiguity blocking and propagation to Defeasible Deontic Logic; (b) we discuss reasoning patterns in the law, especially in criminal law, where we need to deal with both ambiguity blocking and ambiguity propagation in the same legal system and logic; (c) we devise an annotated variant of Defeasible Deontic Logic where we distinguish literals that must be obtained through an ambiguity-blocking mechanism from those that are derived using an ambiguity-propagating mechanism.

CCS CONCEPTS

• Theory of computation → Proof theory; **Automated reasoning**; **• Applied computing → Law**.

KEYWORDS

Defeasible Deontic Logic, Deontic Ambiguities, Ambiguity Blocking, Ambiguity Propagation

ACM Reference Format:
Guido Governatori and Antonino Rotolo. 2023. Deontic Ambiguities in Legal Reasoning. In *Nineteenth International Conference on Artificial Intelligence and Law (ICAIL 2023), June 19–23, 2023, Braga, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3594536.3595175

1 INTRODUCTION

Suppose there is a norm n_1 prescribing C under condition A, and the sanction for not doing C is S_1; at the same time there is a second norm n_2, independent of the first one, forbidding C when B holds; the violation of the second norm is sanctioned by S_2. Moreover, the two norms are at the same level in the hierarchy of the underlying normative system; thus, there are no clear means to asses whether one of the two norms prevails over the other. Here, we say there is a genuine deontic conflict. In other words, we say that the deontic conclusion "C is obligatory" is ambiguous because two chains of reasoning exist, with one supporting this conclusion and another supporting the opposite.

While this scenario is well-known in deontic reasoning, it is less investigated how such a type of conflict can affect further chains of deontic reasoning. Indeed, suppose there is a third norm n_3, independent of n_1 and n_2, mandating D provided that C is permitted. The sanction for the violation of the third norm is S_3.

Consider a case where A and B hold but C and D do not. What are the sanctions (if any) that apply in this situation? What about if n_3 instead of requiring C to be permitted, asks for C to be not forbidden or for being explicitly authorised?

In this paper, we explore this issue and describe different reasoning patterns. In fact, sometimes deontic ambiguities and their effects should be made local and any further reasoning chains should not be prevented, while sometimes all further reasoning chains should be blocked as well. If we consider the scenario consisting of norms n_1–n_3, then we may have the case in Figure 1, where we prevent any effect following from the deontic ambiguity.

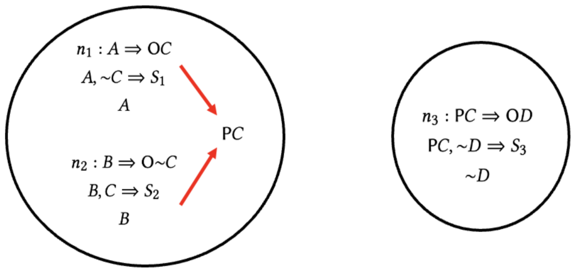
$n_1 : A \Rightarrow OC$
$A, \sim C \Rightarrow S_1$
A
PC
$n_2 : B \Rightarrow O{\sim}C$
$B, C \Rightarrow S_2$
B
$n_3 : PC \Rightarrow OD$
$PC, \sim D \Rightarrow S_3$
$\sim D$

Figure 1: Ambiguity propagation

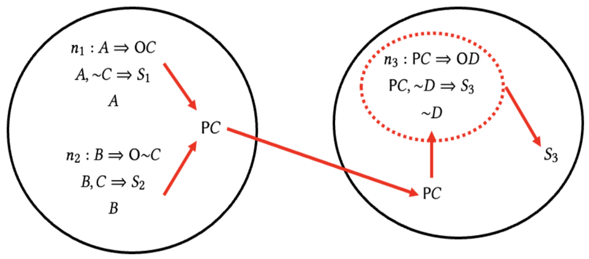
$n_1 : A \Rightarrow OC$
$A, \sim C \Rightarrow S_1$
A
PC
$n_2 : B \Rightarrow O{\sim}C$
$B, C \Rightarrow S_2$
B
$n_3 : PC \Rightarrow OD$
$PC, \sim D \Rightarrow S_3$
$\sim D$
S_3
PC

Figure 2: Ambiguity blocking

On the other hand, we may have the case in Figure 2, where we do not prevent any further effect following from the deontic ambiguity.

This problem has been addressed in propositional Defeasible Logic [15, 3], which is a sceptical approach to non-monotonic reasoning. It is based on a logic programming-like language and is a simple, efficient, but flexible formalism capable of dealing with many different intuitions of non-monotonic reasoning.

Some variants of Defeasible Logic have been proposed, capturing, in particular, the intuitions of different reasoning issues, such as precisely ambiguity blocking and propagation [5]. It was proved that ambiguity propagation corresponds to the grounded semantics, while ambiguity blocking to the semantics for standard Defeasible Logic [9]. However, any formal treatment has yet to be proposed for Defeasible Deontic Logic [10].

The contribution of this paper is threefold:

- We extend the treatment of ambiguity blocking and propagation to Defeasible Deontic Logic;
- We discuss reasoning patterns in the law, especially in criminal law, where we need to deal with both forms of reasoning in the same legal system and logic;
- We devise an annotated variant of Defeasible Deontic Logic where we distinguish literals that must be obtained through an ambiguity-blocking mechanism from those derived using an ambiguity-propagating mechanism.

The layout of the paper is as follows. Section 2 offers an informal introduction to Defeasible Deontic Logic. Section 3 identifies some reasoning patterns, where the distinction between deontic ambiguity propagation and blocking matters, and which are further exemplified in Section 4 and 5 in regard to the principle of legality in criminal law. Section 6 offers a full presentation of the new logic. Section 7 illustrates the logic. A summary concludes the paper.

2 DEFEASIBLE DEONTIC LOGIC IN A NUTSHELL

We assume to work with a rule-based defeasible formalism where rules correspond to norms of legal systems. In particular, Defeasible Deontic Logic (DDL) uses knowledge bases called defeasible theories with three types of elements: (1) facts (denoted by F), (2) rules (denoted by R), and (3) superiority relation $>$. Facts are the input knowledge describing those indisputable things that are true beyond any doubt. Rules are the norms used to obtain (normative) conclusions that are considered plausible, whereas the superiority relation is thought of as a means to establish whether one rule for a conclusion might prevail against another rule for the opposite conclusion.

DDL is standard Defeasible Logic plus the deontic operators O and P which stand, respectively, for obligations and permissions, and the operator $\otimes$ for expressing sanctions or any other obligatory legal countermeasure that follows from violations of obligations. Accordingly, an expression $a \otimes b$ means that a is obligatory. Still, if such an obligation is violated, then the countermeasure b is obligatory (typically, b is a sanction or a compensation of the violation [12]). As often argued in legal theory, norms can be *constitutive* or *prescriptive*. The former are standard (non-deontic) rules, with arrow $\Rightarrow_C$, while the latter ones are deontic rules, such as

$$\alpha\colon a_1, \dots, a_n \Rightarrow_O b \otimes c \qquad\qquad \beta\colon Oc, d_1, \dots, d_m \Rightarrow_P e.$$

If α is applicable (namely, $a_1, \dots, a_n$ are the case), then we derive Ob. Suppose we know $\neg b$, meaning that Ob is violated. In this case, we derive Oc. Accordingly, if we also know that $d_1, \dots, d_m$ are the case, then we conclude that e is permitted, i.e., that Pe. Prescriptive norms thus break down into *obligation norms* and *permissive norms*.

A peculiar feature of this formalism (but similar considerations apply to other logics such as [14]) is that, following [1], it allows for identifying two reasoning patterns for deriving permissions:

- A permission Pa is derived as a *weak permission* because O$\neg a$ is not derivable (in other words, $\neg$O$\neg$ implies P[1]);
- A permission Pa is derived as a *strong permission* because there is an explicit permissive rule (i.e., a rule having the form $b_1, \dots, b_n \Rightarrow_P a$) for it (and the rule is capable to assert its conclusion).

A final feature of DDL is that it supports the notion of *rule conversion*: e.g., given a rule $a_1, \dots, a_n \Rightarrow_C b$, if we derive O$a_1, \dots,$ Oa_n, by conversion we also obtain Ob.

3 DEONTIC AMBIGUITY: SOME REASONING PATTERNS IN THE LAW

First of all, let us define the idea of deontic ambiguity.

Definition 3.1 (Deontic ambiguity). A literal p is *deontically ambiguous* iff there exist two chains of reasoning with one supporting the conclusion Op, while another supporting the conclusion O$\neg p$, and the superiority relation does not resolve this conflict.

We can now consider some reasoning patterns about norms that matter for handling ambiguity.

The following scenario exemplifies the basic and core reasoning pattern. Let $\Box \in \{O, P, C\}$.

Pattern 1.

$$\begin{aligned} &\Rightarrow_O q\\ &\Rightarrow_O \neg q\\ &\quad Pq \Rightarrow_\Box r \end{aligned}$$

A real-life instance to illustrate this pattern is the following. In 2019 Sea Watch 3, an NGO ship carrying migrants rescued at sea, was banned from Italian ports and territorial waters under a policy of the Italian government. After a two-week standoff, the ship contravened the prohibition and entered the port of Lampedusa to complete the rescue mission. As a result, the captain of Sea Watch 3, Ms Carola Rackete, was arrested for violence against public officers. Ms Rackete claimed that the international law of the sea required violating this decree of the Italian government. Suppose that there is no clear way the determine which legal provision should prevail[2]. In addition, if refraining from rescuing is permitted, and if an Italian public authority forces the NGO to rescue, then this is an obstruction of justice and is a criminal offence. The scenario can be reconstructed as follows (we assume that rescuing requires entering the port of Lampedusa):

$$\begin{aligned} r_1\colon &\ \mathit{Endangered\_Migrants}, \mathit{SeaWatch3} \Rightarrow_O \mathit{Rescue}\\ r_2\colon &\ \mathit{Banned\_NGO} \Rightarrow_O \neg\mathit{Rescue}\\ r_3\colon &\ \mathit{Force\_Rescue}, P\neg\mathit{Rescue} \Rightarrow_C \mathit{Offence}\\ r_4\colon &\ \mathit{Offence} \Rightarrow_O \mathit{Imprisonment} \end{aligned}$$

[1]This is guaranteed, e.g., in those deontic logics where the permission is the dual of the obligations: $P =_{def} \neg O \neg$.

[2]We know that a judge subsequently ordered her release on the grounds that the international law of the sea, in fact, required her actions and that international law prevails over the government decree.

Assume that Endangered_Migrants, SeaWatch3, Banned_NGO, Force_Rescue are true. Then the reasoning runs as follows:

(1) *r_1 and r_2 collide and the conflict is not solved, so we can derive both* P*Rescue and* P¬*Rescue;*
(2) *thus r_3 and r_4 are fired, so we have a criminal offence committed by those who force the NGO to rescue.*

A variant of Pattern 1 is the following:[3]

Pattern 2.

$$\begin{aligned} &\Rightarrow_{\mathrm{O}} a\\ &a \Rightarrow_{\mathrm{C}} q\\ &\Rightarrow_{\mathrm{O}} {\sim}q\\ &q \Rightarrow_{\mathrm{C}} r\\ &\Rightarrow_{\mathrm{O}} {\sim}r \end{aligned}$$

Here the deontic ambiguity relates to an unsolved conflict where an obligation would be obtained through a conversion.

A realistic instance illustrating the first three lines of this pattern is the following. Suppose Mary is an illegal immigrant imprisoned in country X. Mary is pregnant and is expected to give birth to her baby in two months in jail. The laws of X state that every baby born in X will get X-citizenship.

$$\begin{aligned} &r_5: \mathit{Imprisoned}, \mathit{Pregnant} \Rightarrow_{\mathrm{O}} \mathit{Birth\_Jail}\\ &r_6: \mathit{Birth\_Jail} \Rightarrow_{\mathrm{C}} \mathit{X\_Born}\\ &r_7: \mathit{X\_Born} \Rightarrow_{\mathrm{C}} \mathit{X\_Citizen}\\ &r_8: \mathit{Illegal\_Immigrant} \Rightarrow_{\mathrm{O}} \neg \mathit{X\_Citizen} \end{aligned}$$

The reasoning runs as follows:

(1) *r_5 proves* O*Birth_Jail;*
(2) *then, by conversion, r_6 and r_7 prove* O*X_Born and X_Citizen, which collides with the conclusion of r_8.*

4 AN APPLICATION: PRINCIPLE OF LEGALITY IN CRIMINAL LAW

Combining different ways of handling deontic ambiguities is paramount, for example, in criminal law.

Consider the principle of legality. It can be summarised as follows (see [4]).

Definition 4.1 (Principle of legality). The following statements define the principle of legality:

(PL1) Criminal offences and penalties must be prescribed only by law. Accordingly, no penalty may be imposed upon any person for committing a criminal offence that did not constitute a criminal offence prior to it being committed and for which a penalty was not prescribed by law.
(PL2) The definition of criminal offence must be strictly construed, and the use of analogy in the interpretation of a criminal offence is prohibited.

As a corollary of the above statements, criminal legality requires that the law is applied in a fair and just manner.

[3]For a literal l we will use $\sim l$ to indicate the complement of l, see Section 6 below for the proper definition.

(PL3) Evidence is assessed through the principle of *in dubio pro reo*, meaning that where there are doubts relating to evidence, the court should adopt the interpretation most favourable to the accused person.
(PL4) If the law was amended on one or more occasions after a criminal offence was perpetrated, the law that most favours the accused must be applied. A penalty heavier than the one applicable at the time a criminal offence was committed may not be imposed upon a person convicted of that offence.

Statements **(PL1)** and **(PL2)** jointly ensure the deontic closure of the criminal legal system—the closure rule *nullum crimen sine lege*—according to which any l which is not prohibited (by criminal law) is permitted (by criminal law) [1, pp. 176–178] (see also [2]). As we have recalled in Section 2, any l can be proved as a weak or strong permission: when any l is proved as permitted through the principle *nullum crimen sine lege*, the literature of deontic logic qualifies l as weakly permitted.

Definition 4.2 (Weak permission and closure of criminal law). Let D be any criminal system. Then, for any l

$$D \nvdash \mathrm{O}l \Rightarrow D \vdash \mathrm{P}{\sim}l. \qquad \textbf{(Weak Permission)}$$

We say that l is *weakly permitted* by the criminal system D.

The deontic closure of D and the idea of weak permission require that l is weakly permitted *precisely because the opposite is not prohibited.*

The interesting case is when statements **(PL1)**–**(PL2)** deontically interplay with **(PL3)**–**(PL4)** as illustrated in Pattern 1. The problem can be in general terms summarised as follows:

- **(PL1)**–**(PL2)** lead to prove that l is weakly permitted each time l is not prohibited;
- If the derivation of Pl is a precondition for the criminal punishment of a crime, should we derive anyway Pl? Indeed, **(PL3)**–**(PL4)** would push for adopting the legal solution which most favours the accused.

The example discussed above shows that a precondition for a criminal offence is that a permission is obtained. What type of permission do we need to derive? Would it be a weak permission enough? In fact, one could argue that weak permission should always be coherent with the *favor rei* principle (see Definition 4.1), thus P¬*Rescue* cannot be a weak permission. Alternatively, one could say that the principle of legality, in the presence of r_3, should simply prevent the derivation of O*Rescue* and O¬*Rescue* and should not support the derivation of P¬*Rescue*.

In general, we may have different alternative scenarios.

Consider this example.

Example 4.3. Consider this case again:

$$\begin{aligned} D = (&\\ &F = \{\neg p\}\\ &R = \{r_9: \Rightarrow_{\mathrm{O}} q,\ r_{10}: \Rightarrow_{\mathrm{O}} \neg q,\ r_{11}: \mathrm{P}q \Rightarrow_{\mathrm{O}} p \otimes s\}\\ &) \end{aligned}$$

Rule r_{11} states that Pq is a precondition for its application and, jointly with F, it leads to a criminal punishment (s as a sanction following from the violation of Op). Since Pq would be derived here as a weak permission on the basis of the principle of legality ((**PL1**)–(**PL2**)), then this deontic conclusion is not acceptable because it would be in contrast with statements (**PL3**)–(**PL4**).

Pattern 3 (Afflictive Weak Permission). *Let D be a criminal system consisting of a set F of facts, a set R of norms, and a superiority relation $>$. If*

- $r: a_1, \dots, a_n \Rightarrow_{\mathrm{O}} c_1 \otimes c_2 \otimes \cdots \otimes c_m \in R$;
- $D \nvdash \mathrm{O}{\sim}p$;
- $D, \mathrm{P}p \vdash \mathrm{O}c_j$, $1 < j \leq m$;
- *If we remove r from D we obtain D' such that* $D' \nvdash \mathrm{O}c_j$;

then $\mathrm{P}p$ *is an* afflictive weak permission *in D and* **Weak Permission** *must not hold.*

On the contrary, if weak permission is not critical for deriving a punishment, then it can be smoothly derived, and all conclusions depending on it could follow as well.

Pattern 4 (Non-afflictive Permission). *Let D be a criminal system consisting of a set F of facts, a set R of norms, and a superiority relation $>$. If* $\mathrm{P}p$ *is not an afflictive weak permission, then* **Weak Permission** *must hold.*

Notice that a non-afflictive permission can be of two types: (1) it is a weak permission, but it is not critical in deriving punishments, (2) it is not a weak permission since it is derived using a permissive norm of the form $a_1, \dots, a_n \Rightarrow_{\mathrm{P}} p$.

5 AMBIGUITY BLOCKING AND AMBIGUITY PROPAGATION: FURTHER REMARKS

Ambiguity blocking and ambiguity propagation are often seen as incompatible facets of non-monotonic reasoning, and they correspond to different semantics. As we have said, an ambiguity occurs when there are two competing rules (rules for conflicting conclusions) and no (clear) mechanisms to resolve the conflict exist. Thus, sceptical non-monotonic formalisms prevent the conclusion of a contradiction, but at the same time, they discard the conclusion of both rules. Hence, none of the two opposite conclusions holds. In other words, there is some ambiguity about which of the two conclusions hold. However, it is possible that the opposite conclusions can be part of the preconditions of other rules. Consider, for example, the scenario where there are two equally compelling different pieces of evidence, one supporting the case that a person was legally responsible for A and the second that the person was not responsible for A. Moreover, if the person was responsible for A, then the person is found guilty. However, according to the presumption of innocence, a person is assumed to be not guilty. The following set of rules can represent this situation:

$$r_1: \mathit{evidence1} \Rightarrow \mathit{responsible}$$
$$r_2: \mathit{evidence2} \Rightarrow \neg\mathit{responsible}$$
$$r_3: \mathit{responsible} \Rightarrow \mathit{guilty}$$
$$r_4: \Rightarrow \neg\mathit{guilty}$$

Where rule r_3 prevails over rule r_4, i.e., $r_3 > r_4$. In this scenario, given the two pieces of evidence, we cannot assert whether *responsible* or $\neg$*responsible* holds; thus, the proposition responsible is ambiguous. A statement is ambiguous when there is an argument supporting it and an argument for its opposite, and there are no means to determine if one of the two arguments is stronger/defeats the other. However, in this situation, we can go on since we cannot assert that responsible holds, but rule r_4 (encoding the so-called presumption of innocence) vacuously holds, and we can conclude $\neg$*guilty*.

However, suppose that, in addition to the conditions stipulated above, if a person was wrongly accused, then the person is entitled to some compensation. This can be encoded by the following two rules:

$$r_5: \neg\mathit{guilty} \Rightarrow \mathit{compensation}$$
$$r_6: \Rightarrow \neg\mathit{compensation}$$

where $r_5 > r_6$. If we continue with our reasoning, rule r_5 is applicable; it defeats r_6, allowing us to establish that "compensation" holds. However, the two pieces of evidence were equally reliable; it does not sound right that the person was wrongly accused. Indeed, it was ambiguous whether the accused was responsible or not. This scenario illustrates that we have to account for two forms of defeasibility: *ambiguity blocking* to block the ambiguity and conclude "not guilty" and *ambiguity propagation* to propagate the ambiguity to compensation, to prevent it from holding. Moreover, [7] argues that these two forms of defeasibility account for different (legal) proof standards: ambiguity propagation to the beyond a reasonable doubt standard and ambiguity blocking to the preponderance of evidence proof standard. The question is whether there are non-monotonic formalisms that can accommodate the two competing (and somehow incompatible) intuitions simultaneously. Variants of Defeasible Logic that accommodate the two intuitions exist. Moreover, [8] proposes a defeasible logic approach where the two variants co-exist and their conclusions can be used simultaneously. In the rest of the paper, we examine how to integrate such an approach to obtain a novel Defeasible Deontic Logic capable of handling both intuitions in a deontic setting to address the issues introduced in Section 4.

6 DEFEASIBLE DEONTIC LOGIC

Defeasible Logic is a constructive non-monotonic formalism. The logic has at its heart a constructive proof theory. The key notion is the notion of derivation. As usual, a derivation is a sequence of expressions either given or derived from previous steps according to the appropriate inference rules (or proof conditions in Defeasible Logic parlance). A characteristic of Defeasible Logic is that the elements of a derivation carry more information; more specifically, each step of a derivation consists of three elements: a logical formula, the indication of how the formula has been derived (either as a positive proof or as a refutation), and the strength of its derivation. The strength and derivation type are encoded in so-called *proof tag*. Notice that the constructive proof theory of the logic allows us to justify and explain every step of a derivation by indicating what rules (corresponding to norms) or facts are required/responsible for the step. Defeasible Deontic Logic extends Defeasible Logic to cover deontic rules and deontic operators and inherits the features of the proof theory, accommodating in it the deontic aspects. For a detailed account of Defeasible Deontic Logic for legal reasoning we refer the reader to [13].

In this paper, we extend the language of Defeasible Deontic Logic by allowing in the language components of the proof theory. Thus formulas can be annotated with what we call proof labels. The idea is that when we have an annotated formula in the antecedent of a rule, the label specifies the standard according to which the formula has to be proved for the rule to be applicable.

We start by defining the language of a defeasible deontic theory.

Let PROP be a set of propositional atoms, and Lab be a set of arbitrary labels (the names of the rules).

Accordingly, PLit = PROP $\cup \{\neg l \mid l \in$ PROP$\}$ is the set of *plain literals*. The set of *deontic literals* ModLit = $\{\Box l, \neg\Box l \mid l \in$ PLit $\wedge$ $\Box \in \{\mathrm{O}, \mathrm{P}\}\}$. Finally, the set of *literals* is Lit = PLit $\cup$ ModLit. The *complement* of a literal l is denoted by $\sim l$: if l is a positive literal p then $\sim l$ is $\neg p$, and if l is a negative literal $\neg p$ then $\sim l$ is p. Note that we will not have specific rules nor modality for prohibitions, as we will treat them according to the standard duality that something is forbidden iff the opposite is obligatory (i.e., $\mathrm{O}\neg p$).

Definition 6.1 (Defeasible Deontic Theory). A *defeasible deontic theory D* is a tuple $(F, R, >)$, where F is the set of facts, R is the set of rules, and $>$ is a binary relation over R (called superiority relation).

Specifically, the set of facts $F \subseteq$ PLit denotes simple pieces of information that are always considered true, like “Sylvester is a cat”, formally $cat(Sylvester)$. In this paper, we subscribe to the distinction between the notions of obligations and permissions, and that of norms, where the norms in the system determine the obligations and permissions in force in a normative system. A Defeasible Deontic Theory is meant to represent a normative system, where the rules encode the norms of the systems, and the set of facts corresponds to a case. As we will see below, the rules are used to conclude the institutional facts, obligations and permissions that hold in a case. Accordingly, we do not admit obligations and permissions as facts of the theory.

The set of rules R contains three *types* of rules: *strict rules*, *defeasible rules*, and *defeaters*. Rules are also of two *kinds*:

- *Constitutive rules* (non-deontic rules) R^{C} model constitutive statements (count-as rules);
- *Deontic rules* to model prescriptive behaviours, which are either *obligation rules* R^{O} which determine when and which obligations are in force, or *permission rules* which represent *strong* (or *explicit*) permissions R^{P}.

Lastly, $> \subseteq R \times R$ is the *superiority* (or *preference*) relation, which is used to solve conflicts in case of potentially conflicting information.

A theory is *finite* if the set of facts and rules are so.

A strict (constitutive) rule is a rule in the classical sense: whenever the premises are indisputable, so is the conclusion. The statement “All computing scientists are humans” is hence formulated through the strict rule[4]

$$CScientist(X) \rightarrow_{\mathrm{C}} human(X),$$

On the other hand, defeasible rules are to conclude statements that can be defeated by contrary evidence. In contrast, defeaters are special rules whose only purpose is to prevent the derivation of the opposite conclusion. Accordingly, we can represent the statement “Computing scientists travel to the city of the conference” through a defeasible rule, whereas “During pandemic travels might be prohibited” through a defeater, like

$$CScientist, PaperAccepted \Rightarrow_{\mathrm{C}} TravelConference$$
$$Pandemic \rightsquigarrow_{\mathrm{C}} \neg TravelConference.$$

On the other hand, a prescriptive behaviour like “At traffic lights it is forbidden to perform a U-turn unless there is a ‘U-turn Permitted’ sign” can be formalised via the general obligation rule

$$AtTrafficLight \Rightarrow_{\mathrm{O}} \neg UTurn$$

and the exception through the permissive rule

$$UTurnSign \Rightarrow_{\mathrm{P}} UTurn.$$

Following the ideas of [12], obligation rules gain more expressiveness with the *sanction operator* $\otimes$ for obligation rules, which is to model sanction chains of obligations. Intuitively, $a \otimes b$ means that a is the primary obligation, but if for some reason we fail to comply with a then b becomes the new obligation in force that sanctions the violation of $\mathrm{O}a$. This operator is used to build chains of preferences called $\otimes$-expressions.

The formation rules for $\otimes$-expressions are: (i) every plain literal is an $\otimes$-expression, (ii) if A is an $\otimes$-expression and b is a plain literal then $A \otimes b$ is an $\otimes$-expression [10].

In general, an $\otimes$-expression has the form ‘$c_1 \otimes c_2 \otimes \cdots \otimes c_m$’, and it appears as consequent of a rule ‘$A(\alpha) \hookrightarrow_{\mathrm{O}} C(\alpha)$’ where $C(\alpha) = c_1 \otimes c_2 \otimes \cdots \otimes c_m$; the meaning of the $\otimes$-expression is: if the rule is allowed to draw its conclusion, then c_1 is the obligation in force, and only when c_1 is violated then c_2 becomes the new in force obligation, and so on for the rest of the elements in the chain. So in this setting, c_m stands for the last chance to comply with the prescriptive behaviour enforced by α, and in case c_m is violated as well, then we will result in a non-compliant situation.

For instance, the previous prohibition to perform a U-turn can foresee a compensatory fine, like

$$AtTrafficLight \Rightarrow_{\mathrm{O}} \neg UTurn \otimes PayFine$$

that has to be paid in case someone does perform an illegal U-turn.

It is worth noticing that we admit $\otimes$-expressions with only one element. The intuition, in this case, is that the obligatory condition does not admit compensatory measures or, in other words, that it is impossible to recover from its violation.

Finally, we introduce the notion of annotated literal [8].

Definition 6.2 (Annotated Literal). Let Π be a set of proof labels. An *annotated literal* is an expression

$$\lambda l$$

where the annotation $\lambda = \pm\mu$ such that the sign $\pm \in \{+, -\}$, and he proof label $\mu \in \Pi$; moreover the literal $l \in$ Lit.

The meaning of an annotated literal is that the literal is provable or refuted with the “strength” indicated by the proof tag. If the sign is +, the literal is provable with strength λ, and if it is −, the literal is refuted with strength λ. Defeasible Deontic Logic is a constructive logic, thus $-\lambda l$ means that it is provable that it is not possible to

[4]Here, we introduce informally the symbols to represent different types of rules, which are formally defined below in Definition 6.3, where $\rightarrow$ denotes a strict rule, $\Rightarrow$ for a defeasible rule, and $\rightsquigarrow$ for a defeated. As usual in Defeasible Logic, see for example, [3], we consider only a propositional version of this logic, and we do not consider function symbols. Every expression with variables represents the finite set of its variable-free instances.

prove l with strength λ. In this paper, we deal with the following set of proof labels:

$$\Pi = \{\Delta, \delta, \partial, \sigma\}$$

The precise meaning of the proof labels will be provided by the formal definitions in the rest of the paper to introduce them in the course of derivations. However, they have the following intuitive meaning:

- Δ: there is a monotonic proof (a proof using strict rules and facts);
- δ: there is a defeasible derivation that propagates ambiguities;
- ∂: there is a defeasible derivation that blocks ambiguities;
- σ: there is a defeasible "credulous" derivation (essentially, ignoring unresolved conflicts).

Defeasible Deontic Logic is a non-monotonic rule-based formalism, and we have already informally introduced the notion of a rule. Formally a rule is defined as below.

Definition 6.3 (Rule). A *rule* is an expression of the form

$$r\colon a_1, \dots, a_n, \lambda_1 b_1, \dots, \lambda_m b_m \hookrightarrow_\Box c,$$

where

(1) $r \in \text{Lab}$ is the unique name of the rule;
(2) $\{a_1, \dots, a_n\}$ is a (possibly empty) set of literals;
(3) $\{\lambda_1 b_1, \dots \lambda_m b_m\}$ is a (possibly empty) set of annotated literals;
(4) An arrow $\hookrightarrow \in \{\Rightarrow, \leadsto\}$ denoting, respectively, defeasible rules, and defeaters;
(5) $\Box \in \{\mathsf{C}, \mathsf{O}, \mathsf{P}\}$;
(6) c, which is either
  (a) a single plain literal $l \in \text{PLit}$, if either (i) $\hookrightarrow \equiv \leadsto$ or (ii) $\Box \in \{\mathsf{C}, \mathsf{P}\}$, or
  (b) an $\otimes$-expression, if $\Box \equiv \mathsf{O}$.

If $\Box = \mathsf{C}$, then the rule is used to derive non-deontic literals (constitutive statements), whilst if $\Box$ is O or P, then the rule is used to derive deontic conclusions (prescriptive statements). $\{a_1, \dots, a_n\} \cup \{\lambda_1 b_1, \dots, \lambda_m b_m\}$ is the antecedent (or body) of r, denoted by $A(r)$. We can split the antecedent of a rule into some subsets: $A^{\mathsf{C}}(r)$ is the set of literals in $A(r)$ not in the scope of a deontic operator; for $\Box \in \{\mathsf{O}, \mathsf{P}\}$ $A^\Box(r)$ is the subset of $A(r)$ of deontic literals, and $A^\lambda(r) = \{\lambda_1 b_1, \dots, \lambda_m b_m\}$ is the set of annotated literal in the antecedent of r. c is the conclusion of the rule, noted as $C(r)$. The conclusion $C(r)$ is a single literal in case $\Box = \{\mathsf{C}, \mathsf{P}\}$; in case $\Box = \mathsf{O}$, then the conclusion is an $\otimes$-expression. Note that $\otimes$-expressions can only occur in prescriptive rules though we do not admit them on defeaters (Condition 6.(a).i), see [10] for a detailed explanation.

We use some abbreviations on sets of rules. The set of strict rules in R is R_s, the set of defeasible rules is R_d, and the set of strict and defeasible rules is R_{sd}. $R^\Box[l]$ is the rule set appearing in R with head l and modality $\Box$, while $R^{\mathsf{O}}[l, i]$ denotes the set of obligation rules where l is the i-th element in the $\otimes$-expression. The abbreviations can be combined.

In the body of a rule, we have two types of literals: "normal" literals and annotated literals. For a rule to be applicable, the normal literals must be provable with whatever conditions are required to prove the conclusion with a given strength. On the contrary, annotated literals specify their required strength independently of what type of conclusion we want to achieve.

Example 6.4. Consider the following rule

$$r\colon a, \mathsf{P}b, +\partial c, -\delta\mathsf{O}\neg c \Rightarrow_{\mathsf{O}} d \otimes \neg e$$

The rule is a defeasible prescriptive rule for d (and $\neg e$); thus the rule is in $R^{\mathsf{O}}_d[d]$ and in $R^{\mathsf{O}}_d[\neg e, 2]$. Moreover, $A(r) = \{a, \mathsf{P}b, +\partial c, -\delta\mathsf{O}\neg c\}$, $A^\Box(r) = \mathsf{P}b, \mathsf{O}\neg c$ and $A^\lambda(r) = \{+\partial c, -\delta\mathsf{O}\neg c\}$. The intuitive meaning of the rule is that: we are able to infer that d is obligatory (and its violation is compensated by the prohibition of e) provided that the normal literals a and $\mathsf{P}b$ hold (according to the normal conditions to infer the conclusion of the rule); in addition, c must be positively proved using the ambiguity blocking standard, but $\mathsf{O}\neg c$ must be refuted with the ambiguity propagation standard.

In addition to the strength of a derivation, we can specify the mode of the derivation. The mode indicates the type of the last rule used to derive the conclusion. As discussed before, we have constitutive rules and regulative rules (prescriptive or obligation and permissive rules). The proof tags extend the proof labels to incorporate this aspect. Accordingly, a proof tag is defined as follows.

Definition 6.5 (Tagged modal formula). Let $\Pi = \{\Delta, \delta, \partial, \sigma\}$ be a set of proof label, and $\{\mathsf{C}, \mathsf{O}, \mathsf{P}\}$ be the set of modalities. A *tagged formula* is an expression of the form

$$\pm\lambda_\Box l$$

where $\pm \in \{+, -\}$, $\lambda \in \Pi$, $\Box \in \{\mathsf{C}, \mathsf{O}, \mathsf{P}\}$, and $l \in \text{PLit}$. The meaning of the tagged literal $\pm\lambda_\Box l$ is:

- $+\lambda_\Box l$: l is *provable* with the strength corresponding to λ and mode $\Box$,
- $-\partial_\Box l$: l is *refuted* with the strength corresponding to λ and mode $\Box$,

Accordingly, the meaning of $+\partial_{\mathsf{O}} p$ is that p is provable as an obligation using the ambiguity blocking proof standard, and $-\delta_{\mathsf{P}}\neg p$ is that we have a refutation for the permission of $\neg p$ using the ambiguity propagating proof standard. Similarly, for the other combinations.

As we will shortly see in the proof conditions and definitions when rules are applicable or discarded, one of the key ideas of Defeasible Deontic Logic is that we use tagged modal formulas to determine what formulas are (defeasibly) provable or rejected given a theory and a set of facts (used as input for the theory). Therefore, when we have asserted a tagged modal formula, let us say $+\partial_{\mathsf{O}} l$ in a derivation (see Definition 6.6 below), we can conclude that the obligation of l ($\mathsf{O}l$) follows from the rules and the facts and that we used a prescriptive rule to derive l and using the conditions blocking ambiguity, given that the proof label is ∂; similarly for permission (using a permissive rule). However, the C modality is silent, meaning that we do not put the literal in the scope of the C modal operator, thus for $+\delta_{\mathsf{C}} l$, the derivation simply asserts that l holds when we propagate the ambiguity (and not that $\mathsf{C}l$ holds, even if the two have the same meaning). For the negative cases (i.e., $-\partial_\Box l$), the interpretation is that it is impossible to derive l with a given mode. Accordingly, we read $-\partial_{\mathsf{O}} l$ as it is impossible to derive l as an obligation. For $\Box \in \{\mathsf{O}, \mathsf{P}\}$, we are allowed to infer $\neg\Box l$, giving a constructive interpretation of the deontic modal operators.

Notice that this is not the case for C, where we cannot assert that $\sim l$ holds (this would require $+\delta_{\mathrm{C}}{\sim}l$); in the logic, failing to prove l does not equate to proving $\neg l$.

We will use the term *conclusions* and tagged modal formulas interchangeably.

The definition of proof is also the standard in DDL.

Definition 6.6 (Proof). Given a defeasible deontic theory D, a proof P of length m in D is a finite sequence $P(1), P(2), \ldots, P(m)$ of tagged modal formulas, where the proof conditions given in the rest of the paper hold.

Given a proof P, $P(1..n)$ denotes the first n steps of P, and we also use the notational convention $D \vdash \pm\lambda_{\Box} l$, meaning that there is a proof P for $\pm\lambda_{\Box} l$ in D.

We now introduce the proof conditions for definite derivations and refutations (corresponding to monotonic forward-looking chaining of rules), sceptical defeasible derivations and refutations, and support an unsupported (essentially corresponding to credulous derivations and refutations ignoring unresolved conflicts).

The definition will be given in a general abstract form that must be instantiated by the appropriate definitions of what it means for a rule to be applicable, supported, discarded and defeated. These definitions depend on the proof tags one wants to characterise.

Definition 6.7 (Definite derivation).
If $P(n+1) = +\Delta_{\Box} p$ then either
(1) $p \in F$ or
(2) there is an applicable strict rule for p.

Definition 6.8 (Definite refutation).
If $P(n+1) = -\Delta_{\Box} p$ then
(1) $p \notin F$ and
(2) all strict rules for p are discarded

Intuitively, the idea behind definite derivations is that there is a chain of (definitely) applicable strict rules, where a rule is definitely applicable where all the normal literals in its body are definitely provable and the annotated literals are provable with the required strengths and modes. Conversely, to refute a literal in the definite sense, we have to show that it is impossible to prove it using only facts and strict rules. Hence, we must have that all rules for that conclusion are definitely discarded. A rule is discarded if one of the normal elements of the body is not definitely provable or at least one of the annotated literals in the body is not provable according to the specification of its annotation.

In addition to the above intuitions, Defeasible Deontic Logic supports the notion of rule conversion. More specifically, a constitutive rule can be used to derive a deontic conclusion (either an obligation or a permission) when (i) the body of the rule is not empty, and (ii) all the literals in the body are not in the scope of deontic operators, and they are provable with the same deontic operator. For example, the constitutive rule

$$r\colon a, b, +\partial c \rightarrow_{\mathrm{C}} d \tag{1}$$

can be used to support the derivation of Od if we can prove a, b and c as obligations.

Formally, the intuitions above are formalised by the following definitions.

Definition 6.9 (Δ-applicable). A strict rule r for p is definitely applicable at step $n+1$ of a derivation P iff
(1) $r \in R^{\Box}[p]$ and $\forall a \in A(r)$:
  (a) if $a \in$ PLit, then $+\Delta_{\mathrm{C}} a \in P(1..n)$;
  (b) if $a = \Box b$, then $+\Delta_{\Box} b \in P(1..n)$;
  (c) if $a = \neg\Box b$, then $-\Delta_{\Box} b \in P(1..)$;
  (d) if $a = \lambda b$, then $\lambda a \in P(1..n)$.
(2) $r \in R^{\mathrm{C}}[p]$, $\Box \in \{\mathrm{O}, \mathrm{P}\}$, $A(r) \neq \emptyset$, $A^{\Box}(r) = \emptyset$ and $\forall a \in A(r)$:
  (a) if $a \in$ PLit, then $+\Delta_{\Box} a \in P(1..)$;
  (b) if $a = \lambda b$, $b \in$ PLit, then $\lambda_{\Box} b \in P(1..n)$.

Accordingly, for the rule in (1) to be Δ-applicable we need to have a proof where we have $+\Delta_{\mathrm{C}} a$, $+\Delta_{\mathrm{C}} b$ and $+\partial_{\mathrm{C}} c$. Moreover, to use the same rule for the derivation of $+\Delta_{\mathrm{O}} c$, we require the derivation of $+\Delta_{\mathrm{O}} a$, $+\Delta_{\mathrm{O}} b$ and $+\partial_{\mathrm{O}} c$.

Definition 6.10 (Δ-discarded). A strict rule r for p is definitely discarded at step $n+1$ of a derivation P iff
(1) $r \notin R^{\Box}[p]$ or $\exists a \in A(r)$:
  (a) if $a \in$ PLit, then $-\Delta_{\mathrm{C}} a \in P(1..n)$;
  (b) if $a = \Box b$, then $-\Delta_{\Box} b \in P(1..n)$;
  (c) if $a = \neg\Box b$, then $+\Delta_{\Box} b \in P(1..)$;
  (d) if $a = \pm\lambda b$, then $\mp\lambda a \in P(1..n)$.
(2) $r \in R^{\mathrm{C}}[p]$, $\Box \in \{\mathrm{O}, \mathrm{P}\}$, and either $A(r) = \emptyset$, $A^{\Box}(r) \neq \emptyset$ or $\exists a \in A(r)$:
  (a) if $a \in$ PLit, then $-\Delta_{\Box} a \in P(1..)$;
  (b) if $a = \pm\lambda b$, $b \in$ PLit, then $\mp\lambda_{\Box} b \in P(1..n)$.

Note that the positive and negative proof conditions (and the notions used inside them) can be obtained from each other by applying the principle of *strong negation* (explained below) to the definition of applicability. The strong negation principle applies the function that simplifies a formula by moving all negations to an innermost position in the resulting formula, replacing the positive tags with the respective negative tags, and the other way around, see [11]. Positive proof tags ensure effective decidable procedures to build proofs; the strong negation principle guarantees that the negative conditions provide a constructive and exhaustive method to verify that deriving the given conclusion is impossible.

Defeasible derivations in Defeasible Deontic Logic have an argumentation like structure, where the first step is to provide an argument (i.e., an applicable rule) for the conclusion we want to prove; then we have to look at all possible attacks to is (i.e., rules for the opposite conclusion); finally, we have to rebut the attacks, either to undercut them or to counterattack with a stronger argument. The schemas for ambiguity blocking and ambiguity propagation have the same structure, but for ambiguity propagating, we make it simpler to attack an argument and harder to rebut. Accordingly, following [5], the proof conditions in Definition 6.11 and 6.12 cover the case for defeasible derivation and refutation for the ambiguity blocking ($\pm\partial$) and ambiguity propagation ($\pm\delta$) variants of the logic. The difference between the two variants is how the notions of applicability, discarded, supported, unsupported and defeated are defined.

Definition 6.11 (Defeasible derivation).
If $P(n+1) = +df_{\Box} p$ then either
(1) $+\Delta_{\Box} p \in P(1..n)$, or

(2) the following three conditions hold
(.1) $-\Delta_{\Box}{\sim}p \in P(1..b)$ and
(.2) there is an applicable strict or defeasible rule for p and
(.3) every rule for ${\sim}p$ is either
(.1) unsupported or
(.2) defeated.

Definition 6.12 (Defeasible refutation).
If $P(n+1) = -df_{\Box}p$ then
(1) $+\Delta_{\Box}p \in P(1..n)$, and
(2) either
(.1) $+\Delta_{\Box}{\sim}p \in P(1..b)$, or
(.2) every strict or defeasible rule for p is discarded, or
(.3) there is a rule for ${\sim}p$ such that
(.1) the rule is supported
(.2) the rule is undefeated.

For the case of $\pm\delta$ we have to introduce additional proof conditions: those for the notion of support.

Definition 6.13 (Support).
If $P(n+1) = +\sigma_{\Box}p$ then either
(1) $+\Delta_{\Box}p \in P(1..n)$ or
(2) there is a supported strict or defeasible rule r for p and for every rule s for ${\sim}p$ either
(.1) s is discarded or
(.2) s is not stronger than r.

Definition 6.14 (Unsupported).
If $P(n+1) = -\sigma_{\Box}p$ then
(1) $-\Delta_{\Box}p \in P(1..n)$ and
(2) for every rule for p either
(.1) the rule is unsupported or
(.2) is defeated by an applicable rule for ${\sim}p$.

What we have to do now is to present the definitions when rules are defeasibly applicable, discarded, supported, unsupported and defeated. As we alluded to above, these notions depend on the variants, and they are different for the characterisation of defeasible derivation in the context of ambiguity blocking and ambiguity propagation. For space and readability reasons, in what follows we limit ourselves to the case of obligations and provide the proof conditions (and accessory definition) for $+\partial_{\mathrm{O}}$ and $+\delta_{\mathrm{O}}$. The negative proof conditions can be obtained from the already discussed principle of strong negation. Similarly, the proof condition for constitutive conclusions and permission can be obtained from the standard definitions given in the literature (see, for instance, [10, 5, 8]) modified according to the mechanism illustrated in the rest of this section. We begin with the case of obligations under ambiguity blocking.

Definition 6.15 ($+\partial_{\mathrm{O}}$).
If $P(n+1) = +\partial_{\mathrm{O}}p$ then either
(1) $+\Delta_{\mathrm{O}}p \in P(1..n)$, or
(2) the following three conditions hold
(.1) $-\Delta_{\mathrm{O}}{\sim}p \in P(1..b)$ and
(.2) $\exists r \in R_{sd}[p]$ such that r is ∂_{O}-applicable for p, and
(.3) $\forall s \in R[{\sim}p]$ either
(.1) s is ∂_{O}-discarded for p or
(.2) s is ∂_{O}-defeated for p.

The first we have to notice is that ambiguity blocking equates unsupported with discarded (where in general, discarded is a stronger notion since it accounts for unresolved conflicts, while unsupported does not).

Definition 6.16 (∂_{O}-applicable). A rule r is ∂_{O}-applicable at step $n+1$ of a derivation P for p iff either
(1) $r \in R^{\mathrm{O}}[p,i]$, $\forall r \in A(r)$,
(a) if $a \in PLit$, then $+\partial_{\mathrm{C}}a \in P(1..n)$,
(b) if $a = \Box b$, then $+\partial_{\Box}b \in P(1..n)$,
(c) if $a = \neg\Box b$, then $-\partial_{\Box}b \in P(1..n)$,
(d) if $a = \lambda b$, then $\lambda b \in P(1..n)$; and
(e) $\forall c_j \in C(r), j < i, +\partial_{\mathrm{O}}c_j \in P(1..n)$ and $+\partial_{\mathrm{C}}{\sim}c_j \in P(1..n)$.
(2) $r \in R^{\mathrm{C}}[p]$, $A(r) \neq \emptyset$, $A^{\Box}(r) = \emptyset$ and $\forall a \in A(r)$:
(a) if $a \in \mathrm{PLit}$, then $+\partial_{\mathrm{O}}a \in P(1..)$;
(b) if $a = \lambda b$, $b \in \mathrm{PLit}$, then $\lambda_{\mathrm{O}}b \in P(1..n)$.

A rule is applicable for an obligation if it is an obligation rule such that all the normal elements in the body are defeasibly provable (using the ambiguity-blocking standard). In addition, for obligation rules, we have to consider the conclusion, which can consist of an $\otimes$-expression. As discussed above, we are allowed to move to the next element of an $\otimes$-expression when the current element is violated. To have a violation, we need (i) the obligation to be in force, and (ii) its content does not hold. $+\partial_{\mathrm{O}}c_i$ indicates that the obligation is in force. For the second part we have two options. The former, $+\partial_{\mathrm{C}}{\sim}c_i$ means that we have "evidence" that the opposite of the content of the obligation holds. The latter would be to have $-\partial_{\mathrm{C}}c_j \in P(1..n)$ corresponding to the intuition that we failed to provide evidence that the obligation has been satisfied. It is worth noting that the former option implies the latter one. For a deeper discussion on the issue, see [6].

The second option to have an applicable rule for an obligation is to use a constitutive rule that converts into an obligation (with the same mechanism described for definite conclusions, with the only difference that normal literals are required to be provable as defeasible obligations).

Definition 6.17 (∂_{O}-discarded). A rule r is ∂_{O}-discarded at step $n+1$ of a derivation P for p iff
(1) $r \in R^{\mathrm{O}}[p,i]$, $\exists r \in A(r)$,
(a) if $a \in PLit$, then $-\partial_{\mathrm{C}}a \in P(1..n)$,
(b) if $a = \Box b$, then $-\partial_{\Box}b \in P(1..n)$,
(c) if $a = \neg\Box b$, then $+\partial_{\Box}b \in P(1..n)$,
(d) if $a = \pm\lambda b$, then $\mp\lambda b \in P(1..n)$; or
(e) $\exists c_j \in C(r), j < i, -\partial_{\mathrm{O}}c_j \in P(1..n)$ and $-\partial_{\mathrm{C}}{\sim}c_j \in P(1..n)$.
(2) $r \in R^{\mathrm{P}}[p]$, $\exists r \in A(r)$,
(a) if $a \in PLit$, then $-\partial_{\mathrm{C}}a \in P(1..n)$,
(b) if $a = \Box b$, then $-\partial_{\Box}b \in P(1..n)$,
(c) if $a = \neg\Box b$, then $+\partial_{\Box}b \in P(1..n)$,
(d) if $a = \pm\lambda b$, then $\mp\lambda b \in P(1..n)$;
(3) $r \in R^{\mathrm{C}}[p]$, and either $A(r) = \emptyset$, or $A^{\Box}(r) \neq \emptyset$ or $\exists a \in A(r)$:
(a) if $a \in \mathrm{PLit}$, then $-\partial_{\mathrm{O}}a \in P(1..)$;
(b) if $a = \mp\lambda b$, $b \in \mathrm{PLit}$, then $\pm\lambda_{\mathrm{O}}b \in P(1..n)$.

The notion of a discarded rule is relevant for the attacking phase of the argumentation like structure. Here, obligation, permission and constitutive rules (converting to obligation) for $\sim p$ must be analysed. Indeed, $\mathrm{O}p$ and $\mathrm{P}{\sim}p$ are in conflict, and the conflict must be resolved. This is why we must incorporate a condition to determine if a permission rule is discarded (not applicable) in this case.

Definition 6.18 (∂_{O}-defeated). A rule r is ∂_{O}-defeated at step $n+1$ of a derivation P for p iff $\exists s \in R[\sim p]$ such that

(1) s is ∂_{O}-applicable for p and
(2) $s > t$.

Finally, only rules capable of producing an obligation can be used in the counterattacking phase. Thus the notion of "defeated" consists of identifying rules able to produce an obligation stronger than non-discarded rules (for an opposing obligation or permission)

We can now move the case of ambiguity propagation. As before, we focus on the proof conditions for obligations. However, in this case, we have to introduce some auxiliary proof conditions, the proof conditions for the notion of support.

Definition 6.19 ($+\delta_{\mathrm{O}}$).
If $P(n+1) = +\delta_{\mathrm{O}}p$ then either
(1) $+\Delta_{\mathrm{O}}p \in P(1..n)$, or
(2) the following three conditions hold
  (.1) $-\Delta_{\mathrm{O}}{\sim}p \in P(1..b)$ and
  (.2) $\exists r \in R_{sd}[p]$ such that r is δ_{O}-applicable for p, and
  (.3) $\forall s \in R[\sim p]$ either
    (.1) s is δ_{O}-unsupported for p or
    (.2) s is δ_{O}-defeated for p.

For ambiguity propagation, the first main difference is condition (2.3.1), where the attacking rules are rebutted if unsupported. While support is a weaker notion than applicable, unsupported is stronger than discarded. Thus, the proof condition makes it easier to attack an argument and harder to rebut it.

Definition 6.20 (Support $+\sigma_{\mathrm{O}}$).
If $P(n+1) = +\sigma_{\mathrm{O}}p$ then either
(1) $+\Delta_{\mathrm{O}}p \in P(1..n)$ or
(2) $\exists r \in R[p]$ such that r is O-supported for every rule s for $\sim p$ either
  (.1) s is discarded or
  (.2) s is not stronger than r

Definition 6.21 (δ_{O}-applicable). A rule r is δ_{O}-applicable at step $n+1$ of a derivation P for p iff either

(1) $r \in R^{\mathrm{O}}[p,i]$, $\forall r \in A(r)$,
  (a) if $a \in PLit$, then $+\delta_{\mathrm{C}}a \in P(1..n)$,
  (b) if $a = \Box b$, then $+\delta_{\Box}b \in P(1..n)$,
  (c) if $a = \neg\Box b$, then $-\delta_{\Box}b \in P(1..n)$,
  (d) if $a = \lambda b$, then $\lambda b \in P(1..n)$; and
  (e) $\forall c_j \in C(r), j < i, +\delta_{\mathrm{O}}c_j \in P(1..n)$ and $+\delta_{\mathrm{C}}{\sim}c_j \in P(1..n)$.
(2) $r \in R^{\mathrm{C}}[p]$, $A(r) \neq \emptyset$, $A^{\Box}(r) = \emptyset$ and $\forall a \in A(r)$:
  (a) if $a \in \mathrm{PLit}$, then $+\delta_{\mathrm{O}}a \in P(1..)$;
  (b) if $a = \lambda b$, $b \in \mathrm{PLit}$, then $\lambda_{\mathrm{O}}b \in P(1..n)$.

The definition of O-supported can be obtained from the definition of δ_{O}-applicable, by replacing the occurrences of δ_{O} with σ_{O}. The intuition behind δ_{O}-applicable and O-supported is essentially the same as that for ∂_{O}-applicable, where the only difference is the strength of the derivation involved.

Definition 6.22 (δ_{O}-discarded). A rule r is δ_{O}-discarded at step $n+1$ of a derivation P for p iff

(1) $r \in R^{\mathrm{O}}[p,i]$, $\exists r \in A(r)$,
  (a) if $a \in PLit$, then $-\delta_{\mathrm{C}}a \in P(1..n)$,
  (b) if $a = \Box b$, then $-\delta_{\Box}b \in P(1..n)$,
  (c) if $a = \neg\Box b$, then $+\delta_{\Box}b \in P(1..n)$,
  (d) if $a = \pm\lambda b$, then $\mp\lambda b \in P(1..n)$; or
  (e) $\exists c_j \in C(r), j < i, -\delta_{\mathrm{O}}c_j \in P(1..n)$ and $-\delta_{\mathrm{C}}{\sim}c_j \in P(1..n)$.
(2) $r \in R^{\mathrm{P}}[p]$, $\exists r \in A(r)$,
  (a) if $a \in PLit$, then $-\delta_{\mathrm{C}}a \in P(1..n)$,
  (b) if $a = \Box b$, then $-\delta_{\Box}b \in P(1..n)$,
  (c) if $a = \neg\Box b$, then $+\delta_{\Box}b \in P(1..n)$,
  (d) if $a = \pm\lambda b$, then $\mp\lambda b \in P(1..n)$;
(3) $r \in R^{\mathrm{C}}[p]$, and either $A(r) = \emptyset$, or $A^{\Box}(r) \neq \emptyset$ or $\exists a \in A(r)$:
  (a) if $a \in \mathrm{PLit}$, then $-\delta_{\mathrm{O}}a \in P(1..)$;
  (b) if $a = \mp\lambda b$, $b \in \mathrm{PLit}$, then $\pm\lambda_{\mathrm{O}}b \in P(1..n)$.

It is easy to verify that the conditions for a rule to be applicable (or discarded) for an obligation under ambiguity propagation are similar to the same conditions (and with the same motivation) under ambiguity blocking. All we have to do is to replace the occurrence of ∂ with δ for normal literals.

Definition 6.23 (δ_{O}-defeated). A rule r is δ_{O}-defeated at step $n+1$ of a derivation P for p iff $\exists s \in R^{[}\sim p]$ such that

(1) s is δ_{O}-applicable for p and
(2) not $t > s$.

Finally, only rules capable of producing an obligation can be used in the counterattacking phase. Thus, as we said, the notion of "defeated" consists of identifying rules that produce an obligation that is not weaker than an attacking rule.

7 DISCUSSION

In the previous section, we defined a logic that requires literals to be derived using both ambiguity blocking and ambiguity propagation. Indeed, with the rule

$$r\colon\ a, \mathrm{P}b, +\partial c, -\delta \mathrm{O}\neg c \Rightarrow_{\mathrm{O}} d \otimes \neg e$$

we are able to infer that d is obligatory (and its violation is sanctioned by the prohibition of e) provided that the normal literals a and $\mathrm{P}b$ hold (according to the normal conditions to infer the conclusion of the rule); in addition, c must be positively proved using the ambiguity blocking standard but $\mathrm{O}\neg c$ must be refuted with the ambiguity propagation standard.

Let us exemplify this idea by elaborating on the scenario presented in Pattern 1.

Example 7.1. Consider the following Defeasible Deontic Theory:

$$F = \{Endangered\_Migrants, SeaWatch3, Banned\_NGO, Force\_Rescue, \neg Pay\_Fine\}$$

$$\begin{aligned} R = \{ & r_1 : Endangered\_Migrants, SeaWatch3 \Rightarrow_O Rescue, \\ & r_2 : Banned\_NGO \Rightarrow_O \neg Rescue, \\ & r_3' : Force\_Rescue, +\delta \mathsf{P}\neg Rescue \Rightarrow_C Offence, \\ & r_4' : Offence \Rightarrow_O Pay\_Fine \otimes Imprisonment, \\ & r_{12} : Storm \Rightarrow_P \neg Rescue, \\ & r_{13} : +\partial Offence, Minor \Rightarrow_O \neg Punishable\} \end{aligned}$$

$$> = \{\langle r_{12}, r_1 \rangle\}$$

Assume the underlying normal standard is ambiguity propagation. Let us modify the case from Pattern 1 as follows:

- r_3 is defined by specifying that $\mathsf{P}\neg Rescue$ can only be derived with ambiguity propagation because, on account of r_4', it an afflictive weak permission (see Pattern 3);
- If $\mathsf{P}\neg Rescue$ is derived from r_{12} (by adding *Storm* to F) we do not have an afflictive weak permission, thus we can derive *Offence* via r_3';
- also notice that r_{13} constitutes a non-afflictive case and, in fact, where we can derive *Offence* with ambiguity blocking standard.

8 SUMMARY

We developed a new Defeasible Deontic Logic able to treat, in the same language and reasoning machinery, different ways in which we handle any genuine deontic conflict—i.e., any deontic ambiguity. This problem is of paramount importance in legal systems, especially in criminal law, where propagating or not an ambiguity matters regarding the application of other norms not directly affected by the conflict. In particular, being more or less sceptical in criminal law is critical to be coherent with the principle of legality and with the deontic closure of the normative system.

Technically, the new logic is a variant of Defeasible Deontic Logic, dealing with both deontic ambiguity blocking and deontic propagation in the same legal system and logic. This means devising an annotated variant of the logic where we distinguish literals that must be obtained through both mechanisms.

An interesting research line for future investigations concerns how to extend this analysis to deontic ambiguities in criminal procedure (we mainly worked on examples of substantive criminal law). As recalled in Section 3 (see [6]), the distinction between ambiguity blocking and propagation is conceptually relevant in procedural law. On the other hand, deontic logic has been proven to be useful to prioritise chains of consecutive actions which has as (final) goal the decision/solution of a conflict [16].

ACKNOWLEDGMENTS

Partially supported by the Project PE01 "Future AI Research" (FAIR, PNRR, CUP: J33C22002830006).

REFERENCES

[1] Carlos E. Alchourrón and Eugenio Bulygin. 1971. *Normative Systems. LEP Library of Exact Philosophy*. Springer Vienna. ISBN: 9783211810194.

[2] Carlos E. Alchourrón and Eugenio Bulygin. 1984. Permission and permissive norms. In *Theorie der Normen*. W. Krawietz et al., (Ed.) Duncker & Humblot.

[3] Grigoris Antoniou, David Billington, Guido Governatori, and Michael J. Maher. 2001. Representation results for defeasible logic. *ACM Transactions on Computational Logic*, 2, 2, 255–287. DOI: 10.1145/371316.371517.

[4] Andrew Ashworth and Jeremy Horder. 2013. *Principles of Criminal Law*. Oxford University Press, Oxford.

[5] David Billington, Grigoris Antoniou, Guido Governatori, and Michael J. Maher. 2010. An inclusion theorem for defeasible logic. *ACM Transactions in Computational Logic*, 12, 1, article 6. DOI: 10.1145/1838552.1838558.

[6] Guido Governatori. 2015. Burden of compliance and burden of violations. In *28th Annual Conference on Legal Knowledge and Information Systems* (Frontieres in Artificial Intelligence and Applications). Antonino Rotolo, (Ed.) IOS Press, Amsterdam, 31–40. DOI: 10.3233/978-1-61499-609-5-31.

[7] Guido Governatori. 2011. On the relationship between Carneades and defeasible logic. In *Fourteenth International Conference on Artificial Intelligence and Law*. Enrico Francesconi and Bart Verheij, (Eds.) ACM, 31–40. DOI: 10.1145/2018358.2018362.

[8] Guido Governatori and Michael J. Maher. 2017. Annotated defeasible logic. *Theory and Practice of Logic Programming*, 17, 5–6, 819–836. DOI: 10.1017/S1471068417000266.

[9] Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. 2004. Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, 14, 5, 675–702. DOI: 10.1093/logcom/14.5.675.

[10] Guido Governatori, Francesco Olivieri, Antonino Rotolo, and Simone Scannapieco. 2013. Computing strong and weak permissions in defeasible logic. *Journal of Philosophical Logic*, 42, 6, 799–829. DOI: 10.1007/s10992-013-9295-1.

[11] Guido Governatori, Vineet Padmanabhan, Antonino Rotolo, and Abdul Sattar. 2009. A defeasible logic for modelling policy-based intentions and motivational attitudes. *Logic Journal of the IGPL*, 17, 3, 227–265. DOI: 10.1093/jigpal/jzp006.

[12] Guido Governatori and Antonino Rotolo. 2006. Logic of violations: a gentzen system for reasoning with contrary-to-duty obligations. *Australasian Journal of Logic*, 4, 193–215. http://ojs.victoria.ac.nz/ajl/article/view/1780.

[13] Guido Governatori, Antonino Rotolo, and Giovanni Sartor. 2021. Logic and the law: philosophical foundations, deontics, and defeasible reasoning. In *Handbook of Deontic Logic and Normative Reasoning*. Vol. 2. Dov M. Gabbay, John Horty, Xavier Parent, Ron van der Meyden, and Leon van der Torre, (Eds.) College Publications, London. Chap. 9, 655–760.

[14] David Makinson and Leendert W. N. van der Torre. 2003. Permission from an input/output perspective. *Journal of Philosophical Logic*, 32, 4, 391–416. DOI: 10.1023/A:1024806529939.

[15] Donald Nute. 1994. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*. Vol. 3. Dov M. Gabbay, C.J. Hogger, and J.A. Robinson, (Eds.), 353–395.

[16] Antonino Rotolo and Clara Smith. 2021. Modelling legal procedures. In *ICAIL'21: Eighteenth International Conference for Artificial Intelligence and Law*. Juliano Maranhão and Adam Zachary Wyner, (Eds.) ACM, 220–224. DOI: 10.1145/3462757.3466089.