Practical Normative Reasoning with Defeasible Deontic Logic

Guido Governatori(✉)

Data61, CSIRO, Dutton Park, Australia
guido.governatori@data61.csiro.au

Abstract. We discuss some essential issues for the formal representation of norms to implement normative reasoning, and we show how to capture those requirements in a computationally oriented formalism, Defeasible Deontic Logic, and we provide the description of this logic, and we illustrate its use to model and reasoning with norms with the help of legal examples.

1 Introduction: Two Normative Reasoning Scenarios

The aim of this contribution is to provide an introduction to a practical computational approach to normative reasoning. To this end we start by proposing two scenarios illustrating some distinctive features of normative reasoning.

Example 1. License for the evaluation of a product [16,26].

Article 1. The Licensor grants the Licensee a license to evaluate the Product.
Article 2. The Licensee must not publish the results of the evaluation of the Product without the approval of the Licensor; the approval must be obtained before the publication. If the Licensee publishes results of the evaluation of the Product without approval from the Licensor, the Licensee has 24 h to remove the material.
Article 3. The Licensee must not publish comments on the evaluation of the Product, unless the Licensee is permitted to publish the results of the evaluation.
Article 4. If the Licensee is commissioned to perform an independent evaluation of the Product, then the Licensee has the obligation to publish the evaluation results.
Article 5. This license terminates automatically if the Licensee breaches this Agreement.

Suppose that the licensee evaluates the product and publishes on her website the results of the evaluation without having received an authorisation from the licensor. Suppose also that the licensee realises that she was not allowed to publish the results of the evaluation, and removes the published results from their website within 24 h from the publication. Is the licensee still able to legally use the product? Since the contract contains a remedial clause (removal within

C. d'Amato and M. Theobald (Eds.): Reasoning Web 2018, LNCS 11078, pp. 1–25, 2018.
https://doi.org/10.1007/978-3-030-00338-8_1

24 h remedies unauthorised publication), it is likely that the license to use the product still holds.

Suppose now, that the licensee, right after publishing the results, posted a tweet about the evaluation of the product and that the tweet counts as commenting on the evaluation. In this case, we have a violation of Article 3, since, even if the results were published, according to Article 2 the publication was not permitted. Thus, she is no longer able to legally use the product under the term of the license.

The final situation we want to analyse is when the publication and the tweet actions take place after the licensee obtained permission for publication. In this case, the licensee has the permission to publish the result and thus they were free to post the tweet. Accordingly, she can continue to use the product under the terms of the license.

Example 2. Consider the following fragment of a contract from the provision of goods and services taken from [10].

3.1 A "Premium Customer" is a customer who has spent more that $10000 in goods.
3.2 Services marked as "special order" are subject to a 5% surcharge. Premium customers are exempt from special order surcharge.
5.2 The (Supplier) shall on receipt of a purchase order for (Services) make them available within one day.
5.3 If for any reason the conditions stated in 4.1 or 4.2 are not met the (Purchaser) is entitled to charge the (Supplier) the rate of $100 for each hour the (Service) is not delivered.

Clause 3.1 provides a definition of Premium Customer for the purpose of the contract. Then Clause 3.2 provides a recipe to compute the price for service marked as special order. In this case, we have two conditions, the standard condition specifying that the price for special order has to be incremented by 5%, and an exception to that computation, when the customer is a Premium Customer. Clause 5.2 establishes an obligation on one of the subject of the norms (a party in the contract, the supplier) based on a condition (that a purchase order had been issued). While we cannot give the complete meaning of Clause 5.3 (it depends on Clauses 4.1 and 4.2, not given here), we can infer that the is triggered in response to a violation of either the conditions specified in 4.1 and 4.2, and that the effect produced is an "entitlement" or in other words a permission.

2 Foundations for Normative Reasoning

The scenarios depicted in the examples illustrate many salient and characteristic aspects of normative reasoning. The first aspect we want to focus on is that when we are given a set of norms normative reasoning is concerned about what are the normative effects that follows from the set of norms, and when such normative effects are in force (or in other terms, when they produce effects that

are "legally" binding). As we will elaborate further in the rest of the paper, we will distinguish between two types of "normative" effects. The first type is that some norms (for example Article 3.1 and Article 3.2 in Example 2) specify the meaning of terms/concepts in the context where the norms are valid. The second type of effects determine constraints affecting the subjects of the norms (essentially all other Articles in the two examples above). In both cases, the norms consist of two parts the normative effects, and the conditions under which the normative effects are in force (and when they are in force), for example, taking Article 2 of Example 1 the normative effect is the prohibition of publishing the result of the evaluation of a product, and the condition is not having the authorisation from the licensor to publish. Accordingly, norms can be represented as *if*... *then*... rules, where the *if* part determines the conditions of applicability of the norm, and the *then* part gives the normative effects.

Based on the discussion so far, we can consider a normative system as a set of clauses (norms), where the clauses/norms are represented as *if*... *then* rules. Every clause/norm is represented by one (or more) rule(s) with the following form:

$$A_1, \ldots, A_n \hookrightarrow C \tag{1}$$

where $A_1, \ldots, A_n$ and the conditions of applicability of the norm and C is the "effect" of the norm. According to the type of effect we can classify the norms/rules as

- constitutive (also known as counts-as) rules that define the terms used in the normative systems, or in other terms they create "institutional facts" from brute facts and other brute facts.[1]
- prescriptive rules, that determine what "normative" effects are in force based on the conditions of applicability.

For normative effects we consider:

- Obligation,
- Prohibition,
- Permission.

These notions can be defined as follows [30]:

Obligation: a state, an act, or a course of action to which a Bearer is legally bound, and which, if it is not achieved or performed, results in a Violation.

Prohibition: a state, an act, or a course of action to which a Bearer is legally bound, and which, if it is achieved or performed, results in a Violation.

Permission: something is permitted if a Bearer has no Prohibition or Obligation to the contrary. A weak Permission is the absence of the Prohibition or Obligation to the contrary; a strong Permission is an exception or derogation of the Prohibition or Obligation to the contrary.

[1] For extended discussions on constitutive or counts-as rules, see the seminal work by Searle [33] and, for formal treatments, the comprehensive [25].

One of the functions of norms is to regulate the behaviour of their subjects by imposing constraints on what the subjects can or cannot do, what situations are deemed legal, and which ones are considered to be illegal. There is an important difference between the constraints imposed by norms and other types of constraints. Typically, a constraint means that the situation described by the constraint cannot occur. For example, suppose you have a constraint A. This means that if $\neg A$ (the negation of A, that is, the opposite of A) occurs, then we have a contradiction, or in other terms, we have an impossible situation. Norms, on the other hand, can be violate $\neg A$, we do not have a contradiction, but rather a violation, or in other terms we have a situation that is classified as "illegal". From a logical point of view, we cannot represent the constraint imposed by a norm simply by A, since the conjunction of A and $\neg A$ is a contradiction. Thus, we need a mechanism to identify the constraints imposed by norms. This mechanism is provided by modal (deontic) operators.

2.1 Deontic Operators

Here we are going to consider the following deontic operators: O for obligation, F for prohibition (forbidden), and P for permission. The deontic operators are modal operators. A modal operator applies to a proposition to create a new proposition where the modal operator qualifies the "truth" of the proposition the operator is applied to. Consider, for instance, based on Example 1, the proposition *publishEvaluation* meaning that "the licensee published the results of the evaluation of the product". We can distinguish the following propositions/statements:

- *publishEvaluation*: this is a factual statement that is true if the licensee has published the evaluation (of the product), and false otherwise (in this case $\neg$*publishEvaluation* is true).
- O*publishEvaluation*: this is a deontic statement meaning that the licensee has the obligation to publish the evaluation of the product. The statement is true, if the obligation to publish the evaluation is in force in the particular case.
- F*publishEvaluation*: this is a deontic statement meaning that the licensee has the prohibition to publish the evaluation of the product. The statement is true, if the prohibition to evaluate the produce is in force in the particular case.
- P*publishEvaluation*: this is a deontic statement meaning that the licensee has the permission to publish the evaluation of the product. The statement can be evaluated as true, if the permission to publish the evaluation is in force in the particular case.

Looking again at Example 1, we have that the prohibition to publish the evaluation of the product is in force (thus, F*publishEvaluation* is true) if the licensee does not get the approval for publication from the licensor, or if she is not commissioned to evaluate the product. However, if she gets the approval (Article 2), then the permission to publish is in force (P*publishEvaluation* holds), and if she is commissioned, then she is under the obligation to publish the evaluation.

Now the question is if she is commissioned to evaluate the product (and thus have the obligation to publish the evaluation), then does she have the permission to publish the evaluation? To provide an answer to this question, we go back to the idea that norms define constraints on what is legal and what is illegal, and how to model a violation. Obligations and prohibitions can be violated; according to some legal scholars, the possibility of being violated can be used to define an obligation (prohibition). A violation means that the content of the obligation has not been met. As we have alluded to above, it is important to notice that a violation does not result in an inconsistency. A violation is, basically, a situation where we have[2]

$$\mathsf{O}A \text{ and } \neg A, \text{ or } \mathsf{F}A \text{ and } A. \tag{2}$$

On the other hand, a permission, as we have already said, is the lack of the obligation to the contrary (and a violation requires an obligation it violates), thus permissions cannot be violated. Now, going back to the question, suppose that we have the obligation to publish the evaluation but we do not have the corresponding permission; this means that we have the prohibition of publishing the evaluation. Thus, we have at the same time the obligation and the prohibition of publishing the evaluation. We have two cases, we publish the evaluation, complying with the obligation to publish, but we violated the prohibition; we refrain from publishing the evaluation complying with prohibition (lack of permission), this violates the obligation to publish. In both cases, we get a violation, while, the intuition is that if we have the obligation to publish, and we do publish, we are fully compliant with the norms. Thus, we can conclude that, in case, we have an obligation, we should have (or derive) the corresponding permission.

From the definitions of the normative effects and the discussion above, we get the following equivalences:

$$\mathsf{F}A \equiv \mathsf{O}\neg A \qquad \mathsf{O}A \equiv \mathsf{F}\neg A \qquad \mathsf{P}A \equiv \neg\mathsf{O}\neg A \tag{3}$$

The first two equivalences state that a prohibition corresponds to a negative obligation, and that a permission is the lack of the obligation to the contrary, or more precisely, that the obligation to the contrary is not in force. However, for permission, we can distinguish two notions: *strong permission*, which specifies that the obligation to the contrary is not in force because it has been derogated by the explicit permission; and *weak permission* simply meaning that the obligation to the contrary is not in force (this can be because the conditions of applicability for the obligation to the contrary do not hold, or simply because there are not norms for it). We will use P_w to indicate weak permission, P_s for strong permission and P where the strength of the permission is not specified. We further assume that, for every proposition A, $\mathsf{O}A \rightarrow \mathsf{P}_\mathsf{s}A$,[3] and $\mathsf{P}_\mathsf{s}A \rightarrow \mathsf{P}_\mathsf{w}A$

[2] In (modal/deontic) logics where $\mathsf{O}A$ does not imply A a violation does not imply a contradiction, thus there are consistent state to represent situations corresponding to violations.

[3] In this section we use "$\rightarrow$" to indicate material implication of classical propositional logic. In the rest of the paper it will be used to denote a particular type of rules in Defeasible Logic.

and that the equivalence $\neg \mathsf{O} \neg A \equiv \mathsf{P}_{\mathsf{w}} A$ for weak permission but not for strong permission[4].

2.2 Handling Violations

Given that a violation is not a contradiction, but a situation where we have $\mathsf{O}A$ and $\neg A$, then we can reason about violations, and having provisions depending on violations. An important issue is related to the so called *contrary-to-duty obligations* (CTD)[5]. Shortly a contrary-to-duty obligation is an obligation (or more generally a normative effect) that enters in force because another obligation has been violated. The general structure of a CTD is

1. an obligation $\mathsf{O}A$, and
2. a norm such that the opposite of the content of the obligation is part of the conditions of applicability of the norm, so something with the following structure $C_1, \ldots, C_n, \neg A \hookrightarrow \mathsf{O}B$

When the obligation $\mathsf{O}A$ is in force, and the conditions of applicability of the norm holds, we have (1) a violation, and (2) the obligation $\mathsf{O}B$ enters in force. However, if $C_1, \ldots, C_n$ do no contain $\mathsf{O}A$, then the two provisions are somehow independent, and the CTD is somehow accidental. Nevertheless, normative systems, typically contain prescriptions related to violations, to establish penalties to compensate for the violation (an instance of this is Article 2 in Example 1); or, in some other cases, they contain provisions for ancillary penalties or normative effects that, per se, do not compensate for the violation (this is the case of clause 5.3 in Example 2). In both cases they should make explicit the relationship with a violation. Accordingly, following [14,21] we introduce the operator ($\otimes$) proposed in [21] for CTD and reserve it for *compensatory obligations*, and we distinguish between norms

$$C_1, \ldots, C_n, \mathsf{O}A, \neg A \hookrightarrow \mathsf{O}B \tag{4}$$

$$C_1, \ldots, C_n \hookrightarrow A \otimes B \tag{5}$$

A norm with the form as in 4 means that the obligation of B requires to have a violation of A as its conditions to be in force. While $A \otimes B$ means that $\mathsf{O}A$ is in force, but if violated (i.e., $\neg A$ holds) then $\mathsf{O}B$ is in force, and the fulfilling the obligation of B compensates for the violation of the obligation of A.

Notice that not all violations are compensable, this means that there are situations, that while logically consistent, are not legal according to a normative system, and normative systems can contain provisions about this type of situations, see the termination clause, Article 5 in Example 1. An instance of such a case, is when the licensee is commissioned to evaluate the produce, but she

[4] Suppose that you do not have an explicit permission for $\neg A$; this means that there are no rules to derogate the obligation of A, but it does not mean that there are no norms mandating A.

[5] The term contrary-to-duty was coined by Chisholm [6]; his work inspired a wealth of research on deontic logic to address the paradox proposed in [6] and other CTD paradoxes. For overviews of such paradoxes and development in the field see [5,31].

does not (Article 4). Another instance, is when the licensee publishes the results of the evaluation without approval, and fails to remove within the allotted time (Article 2).

2.3 Defeasibility: Handling Exceptions

So far we have assumed that norms are represented by normative conditionals with the form $A_1, \ldots, A_n \hookrightarrow C$, but we have not discussed the nature of such conditionals, which are generally defeasible and do not correspond to the if-then material implication of classical propositional logic. Norms are meant to provide general principles, but at the same time they can express exceptions to the principle. It is well understood in Legal Theory and Artificial Intelligence and Law [8,32] that, typically, there are different types of "normative conditionals", but in general normative conditionals are defeasible. Defeasibility is the property that a conclusion is open in principle to revision in case more evidence to the contrary is provided, the conclusion is obtained using only the information available when the conclusion is derived. Defeasible reasoning is in contrast to monotonic reasoning of propositional logic, where no revision is possible. In addition, defeasible reasoning allows reasoning in the face of contradictions, which gives rise to ex false quodlibet in propositional logic. One application of defeasible reasoning is the ability to model exceptions in a simple and natural way.

The first use of defeasible rules is to capture conflicting rules/norms without making the resulting set of rules inconsistent. The following two rules conclude with the negation of each other

$$A_1, \ldots, A_n \hookrightarrow C \tag{6}$$

$$B_1, \ldots, B_m \hookrightarrow \neg C \tag{7}$$

without defeasibile rules, rules with conclusions that are negations of each other could give rise, should $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ both hold, to a contradiction, i.e., C and $\neg C$, and consequently *ex falso quodlibet*. Instead, defeasible reasoning is sceptical; that is, in case of a conflict such as the above, it refrains from taking any of the two conclusions, unless there are mechanisms to solve the conflict.

Norms, typically give the general conditions of applicability to guarantee that some particular conclusions hold, and then list the possible exceptions (Articles 2, 3 in Example 1 and Clause 3.2 in Example 2 provide explicit instances of this phenomenon, while the exception is implicit in Article 1, it assumes an implicit norm that forbids the use of the product without a license). Exceptions limit the applicability of basic norms/rules. The general structure to model exceptions is given by the following rules:

$$A_1, \ldots, A_n \hookrightarrow C \tag{8}$$

$$A_1, \ldots, A_n, E_1 \hookrightarrow \neg C \tag{9}$$

$$\ldots$$

$$A_1, \ldots, A_n, E_m \hookrightarrow \neg C \tag{10}$$

In this case, the rules from is more specific than the first, and thus it forms an exception to the first, i.e., a case where the rule has extra conditions. In a classical logic setting this would result in contradiction as soon as the conditions of applicability of the general norm/rule (8) and the any of the exceptions (E_i) holds. In defeasible reasoning the extra condition encodes the exception, and blocking the conclusion of the first rule. It is possible to use classical (monotonic) reasoning by encoding (8) as

$$A_1 \wedge \cdots \wedge A_n \wedge \bigwedge_{i=1}^{m} \neg E_i \rightarrow C \tag{11}$$

The above representation still suffers from several drawbacks (for a detailed discussion see [20]). To apply this rule, we have to know for each E_i whether it is true or false. This means that if in a (legal) case we only have partial knowledge of the facts of the cases we cannot draw conclusions, and in the extreme case this approach would require factually omniscient knowledge. The second drawback depends on the possibility of having exceptions to the exceptions, thus for each exception E_i, we could have rules

$$A_1, \ldots, A_n, E_i, F_1^i \hookrightarrow C \tag{12}$$

$$\ldots$$

$$A_1, \ldots, A_n, E_i, F_k^i \hookrightarrow C \tag{13}$$

To capture exceptions to the exceptions we have to revise (11) to

$$A_1 \wedge \ldots \wedge A_n \wedge \bigwedge_{i=1}^{m} \left(\neg E_i \wedge \bigvee_{j=1}^{k} F_j^i\right) \rightarrow C \tag{14}$$

with the same problems of factual omniscience as above (and recursive refinement in case of exceptions to the exceptions to the exceptions). In contrast in defeasible reasoning, exceptions can simply be represented by replacing rules (9)–(10) with

$$E_1 \hookrightarrow \neg C \tag{15}$$

$$\ldots$$

$$E_m \hookrightarrow \neg C \tag{16}$$

plus a mechanism to solve conflict. The key point is that unless one is able to conclude the conditions of applicability of a rule, the rule does not fire, and it is not able to produce its conclusion.

In this section we gave a short overview of the salient features for practical legal reasoning, in the next section we are going to present how to implement them in a computationally oriented formalism: Defeasible Deontic Logic.

3 Defeasible Deontic Logic

Defeasible Deontic Logic is an extension of Defeasible Logic with deontic operators, and the operators for compensatory obligation introduced in [21]. Defeasible Logic is a simple rule based computationally oriented (and efficient) nonmonotonic formalism, designed for handling exception in a natural way. While Defeasible Logic was originally proposed by Nute [29], we follow the formalisation proposed in [2]. Defeasible Logic is a constructive logic with its proof theory and inference condition as its core. The logic exploits both positive proofs, a conclusion has been constructively prove using the given rules and inference conditions (also called proof conditions), and negative proofs: showing a constructive and systematic failure of reaching particular conclusions, or in other terms, constructive refutations. The logic uses a simple language, that proved successful in many application area, due to the scalability of the logic, and its constructiveness. These elements are extremely important for normative reasoning, where an answer to a verdict is often nor enough, and full traceability is needed. This feature is provided by the constructive proof theory of Defeasible logic.

In the rest of this section we first provide an informal presentation of basic Defeasible Logic, we informally discuss how the extend it to cover the normative features we outlined in the previous section, and then finally, we provide the formal presentation of Defeasible Deontic Logic.

3.1 Basic Defeasible Logic

Knowledge in Defeasible logic is structured in three components:

- A set of facts (corresponding to indisputable statements represented as literals, where a literal is either an atomic proposition or its negation).
- A set of rules. A rule establishes a connection between a set of premises and a conclusion. In particular, for reasoning with norms, it is reasonable to assume that a rule provides the formal representation of a norm. Accordingly, the premises encode the conditions under which the norm is applicable, and the conclusion is the normative effect of the norm.
- A preference relation over the rules. The preference relation just gives the relative strength of rules. It is used in contexts where two rules with opposite conclusions fire simultaneously, and determines that one rule overrides the other in that particular context.

Formally, the knowledge in the logic is organised in Defeasible Theories, where a Defeasible Theory D is a structure

$$(F, R, \prec) \tag{17}$$

where F is the set of facts, R is the set of rules, and $\prec$ is a binary relation over the set of rules, i.e., $\prec \subseteq R \times R$.[6]

[6] Defeasible Logic does not impose any property for $\prec$. However, in many applications it is useful to assume that the transitive closure to be acyclic to prevent situations where, at the same time a rule overrules another rule and it is overridden by it.

As we have alluded to above, a rule is formally a binary relation between, a set premises and a conclusion. Thus, if Lit is the set of literals, the set Rule of all rules is:

$$\text{Rule} \subseteq 2^{\text{Lit}} \times \text{Lit}. \tag{18}$$

Accordingly, a rule is an expression with the following form:[7]

$$r\colon a_1, \ldots, a_n \hookrightarrow c \tag{19}$$

where r is a unique label identifying the rule. Given that a rule is a relation, we can ask what is the strength of the link between the premises and the conclusion. We can distinguish three different strengths: (i) given the premises the conclusion always holds, (ii) given the premises the conclusion holds sometimes, and (iii) given the premises the opposite of the conclusions does not hold. Therefore, to capture theses types Defeasible Logic is equipped with three types of rules: *strict rules*, *defeasible rules* and *defeaters*. We will use $\rightarrow$, $\Rightarrow$ and $\rightsquigarrow$ instead of $\hookrightarrow$ to represent, respectively, strict rules, defeasible rules and defeaters.

Given a rule like rule r in (19) we use the following notation to refer to the various elements of the rule. $A(r)$ denotes the *antecedent* or *premises* of the rule, in this case, $\{a_1, \ldots, a_n\}$, and $C(r)$ denotes the *conclusion* or *consequent*, that is, c. From time to time we use *head* and *body* of a rule to refer, respectively, to the consequent and to the antecedent of the rule.

Strict rules are rules in the classic sense: whenever the premises are indisputable so is the conclusion. Strict rules can be used to model legal definitions that do not admit exceptions, for example the definition of minor: "'minor' means any person under the age of eighteen years". This definition can be represented as

$$age(x) < 18yrs \rightarrow minor(x). \tag{20}$$

Defeasible Rules are rules such that the conclusions normally or typically follow from the premises, unless there are evidence or reasons to the contrary.

Defeaters are rules that do not support directly the derivation of a conclusion, but that can be used to prevent a conclusion.

Finally, for the *superiority relation*, given two rules r and s, we use $r \prec s$ to indicate that rule r defeats rule s; in other terms, if the two rules are in conflict with each other and they are both applicable, then r prevails over s, and we derive only the conclusion of r.

Example 3. We illustrate defeasible rules and defeaters with the help of the definition of complaint from the Australian Telecommunication Consumer Protections Code 2012 TCP-C268_2012 May 2012 (TCPC).

> **Complaint** means an expression of dissatisfaction made to a Supplier in relation to its Telecommunications Products or the complaints handling process itself, where a response or Resolution is explicitly or implicitly expected by the Consumer.

[7] More correctly, we should use $r\colon \{a_1, \ldots, a_n\} \hookrightarrow c$. However, to improve readability, we drop the set notation for the antecedent of rule.

> An initial call to a provider to request a service or information or to request support is not necessarily a Complaint. An initial call to report a fault or service difficulty is not a Complaint. However, if a Customer advises that they want this initial call treated as a Complaint, the Supplier will also treat this initial call as a Complaint.
> If a Supplier is uncertain, a Supplier must ask a Customer if they wish to make a Complaint and must rely on the Customer's response.

Here is a (simplified) formal representation:

$$tcpc_1\colon \mathit{ExpressionDissatisfaction} \Rightarrow \mathit{Complaint}$$
$$tcpc_2\colon \mathit{InformationCall} \Rightarrow \neg \mathit{Complaint}$$
$$tcpc_3\colon \mathit{ProblemCall}, \mathit{FirstCall} \rightsquigarrow \mathit{Complaint}$$
$$tcpc_4\colon \mathit{AdviseComplaint} \Rightarrow \mathit{Complaint}$$

where $tcpc_2 \prec tcpc_1$ and $tcpc_4 \prec tcpc_2$.

The first rule $tcpc_1$ sets the basic conditions for something to be a complaint. On the other hand, rule $tcpc_2$ provides an exception to the first rule, and rule $tcpc_4$ is an exception to the exception provided by rule $tcpc_2$. Finally, $tcpc_3$ does not alone warrant the call to be a complaint (though, it does not preclude the possibility that the call turns out to be a complaint; hence the use of a defeater to capture this case).

Defeasible Logic is a constructive logic. This means that at the heart of it we have its proof theory, and for every conclusion we draw from a defeasible theory we can provide a proof for it, giving the steps used to reach the conclusion, and at the same time, providing a (formal) explanation or justification of the conclusion. Furthermore, the logic distinguishes *positive* and *negative* conclusion, and the strength of a conclusion. This is achieved by labelling each step in a derivation with a proof tag. As usual a derivation is a (finite) sequence of formulas, each obtained from the previous ones using inference conditions.

Let D be a Defeasible Theory. The following are the proof tags we consider for basic Defeasible Logic:

- $+\Delta$ if a literal p is tagged by $+\Delta$, then this means that p is provable using only the facts and strict rules in a defeasible theory. We also say that p is *definitely provable* from D.
- $-\Delta$ if a literal p is tagged by $-\Delta$, then this means that p is refuted using only the facts and strict rules in a defeasible theory. In other terms, it indicates that the literal p cannot be proved from D using only facts and strict rules. We also say that p is *definitely refuted* from D.
- $+\partial$ if a literal p is tagged by $+\partial$, then this means that p is *defeasibly provable* from D.
- $-\partial$ if a literal p is tagged by $-\partial$, then this means that p is *defeasibly refutable* from D.

Some more notation is needed before explaining how tagged conclusions can be asserted. Given a set of rules R, we use R_x to indicate particular subsets of rules: R_s for strict rules, R_d for defeasible rules, R_{sd} for strict or defeasible rules, R_{dft} for defeaters; finally $R[q]$ denotes the rules in R whose conclusion is q.

There are two ways to prove $+\Delta p$ at the n-th step of a derivation: the first is that p is one of the facts of the theory. The second case is when we have a strict rule r for p and all elements in the antecedent of r have been definitely proved at previous steps of the derivation.

For $-\Delta p$ we have to argue that there is no possible way to derive p using facts and strict rules. Accordingly, p must not be one of the facts of the theory, and second for every rule in $R_s[p]$ (all strict rules which are able to conclude p) the rule cannot be applied, meaning that at least one of the elements in the antecedent of the rule has already refuted (definitely refuted). The base case is where the literal to be refuted is not a fact and there are no strict rules having the literal as their head.

Defeasible derivations have a three phases argumentation-like structure[8]. To show that $+\partial p$ is provable at step n of a derivation we have to:[9]

1. give an argument for p;
2. consider all counterarguments for p; and
3. rebut each counterargument by either:
   (a) showing that the counterargument is not valid;
   (b) providing a valid argument for p defeating the counterargument.

In this context, in the first phase, an argument is simply a strict or defeasible rule for the conclusion we want to prove, where all the elements are at least defeasibly provable. In the second phase we consider all rules for the opposite or complement of the conclusion to be proved. Here, an argument (counterargument) is not valid if the argument is not supported.[10] Here "supported" means that all the elements of the body are at least defeasibly provable.

Finally to defeasibly refute a literal, we have to show that either, the opposite is at least defeasible provable, or show that an exhaustive search for a constructive proof for the literal fails (i.e., there are rules for such a conclusion or all rules

[8] The relationships between Defeasible Logic and argumentation are, in fact, deeper than the similarity of the argumentation like proof theory. [18] prove characterisation theorems for defeasible logic variants and Dung style argumentation semantics [7]. In addition, [11] proved that the Carneades argumentation framework [9], widely discussed in the AI and Law literature, turns out to be just a syntactic variant of Defeasible Logic.

[9] Here we concentrate on proper defeasible derivations. In addition we notice that a defeasible derivations inherit from definite derivations, thus we can assert $+\partial p$ if we have already established $+\Delta p$.

[10] It is possible to give different definitions of support to obtain variants of the logic tailored for various intuitions of non-monotonic reasoning. [4] show how to modify the notion of support to obtain variants capturing such intuitions, for example by weakening the requirements for a rule to be supported: instead of being defeasibly provable a rule is supported if it is possible to build a reasoning chain from the facts ignoring rules for the complements.

are either "invalid" argument or they are not stronger than valid arguments for the opposite).

Example 4. Consider again the set of rules encoding the TCPC 2012 definition of complaint given in Example 3. Assume to have a situation where there is an initial call from a customer who is dissatisfied with some aspects of the service received so far where she asks for some information about the service. In this case rules $tcpc_1$ and $tcpc_2$ are both applicable (we assume that the facts of the case include the union of the premises of the two rules, but *AdviseComplaint* is not a fact). Here, $tcpc_2$ defeats $tcpc_1$, and $tcpc_4$ cannot be used. Hence, we can conclude $-\partial AdviseComplaint$ and consequently $+\partial\neg Complaint$ and $-\partial Complaint$. However, if the customer stated that she wanted to complain for the service, then the fact *AdviseComplaint* would appears in the facts. Therefore, we can conclude $+\partial AdviseComplaint$, making then rule $tcpc_4$ applicable, and we can reverse the conclusions, namely: $+\partial Complaint$ and $-\partial\neg Complaint$.

While the Defeasible Logic we outlined in this section and its variants are able to model different features of legal reasoning (e.g., burden of proof [24] and proof standards [11] covering and extending the proof standards discussed in [9]), we believe that a few important characteristics of legal reasoning are missing. First, we do not address the temporal dimension of norms (and, obviously, this is of paramount importance to model norm dynamics), and second, we do not handle the normative character of norms: norms specify what are the obligations, prohibitions and permissions in force and what are the conditions under which they are in force. In the next sections we are going to extend Defeasible Logic with (1) deontic operators, to capture the normative nature of norms and (2) time, to model the temporal dimensions used in reasoning with norms.

3.2 The Intuitions Behind Defeasible Deontic Logic

In the language of Defeasible Deontic Logic the set of literals Lit is partitioned in *plain literals* and *deontic literals*. A plain literal is a literal in the sense of basic defeasible logic, while a deontic literal is obtained by placing a plain literal in the scope of a deontic operator or a negated deontic operator. Accordingly, expressions like $\mathsf{O}l$, $\neg\mathsf{P}l$ and $\mathsf{F}\neg l$ are deontic literals, where l is plain literal. We use ModLit to indicate the set of all deontic literals. We have now to give the construction for the operator for compensatory obligations ($\otimes$). We will refer to such expression as $\otimes$-expressions, whose construction rules are as follows:

(a) every literal $l \in$ Lit is an $\otimes$-expression;
(b) if $l_1, \ldots, l_n$ are plain literals, then $l_1 \otimes \cdots \otimes l_n$ is an $\otimes$-expression;
(c) nothing else is an $\otimes$-expression;
(d) the set of all $\otimes$-expression is denoted by Oexpr.

The main idea is that instead of a single family or rules, we are going to have to have two families: one for constitutive rules, and the other for prescriptive rules. The behaviour of these rules is different. Given the constitutive defeasible rule

$$a_1, \ldots, a_n \Rightarrow_C b \tag{21}$$

we can assert b, given $a_1, \ldots, a_n$, thus the behaviour of constitutive rule is just the normal behaviour of rules we examined in the previous section. Thus, when the rule is applicable, we can derive $+\partial_C b$ meaning that b is derivable. For prescriptive rules the behaviour is a different. From the rule

$$a_1, \ldots, a_n \Rightarrow_{\mathsf{O}} b \tag{22}$$

we conclude $+\partial_{\mathsf{O}} b$ meaning that we derive $\mathsf{O}b$ when we have $a_1, \ldots, a_n$. Thus we conclude the obligation of the consequent of the rule, namely that b is obligatory, i.e., $\mathsf{O}b$, not just the consequent of the rule, i.e., b.[11]. Furthermore, for prescriptive rules we can have rules like

$$a_1, \ldots, a_n \Rightarrow_{\mathsf{O}} b_1 \otimes \cdots \otimes b_m \tag{23}$$

In this case, when the rule is applicable, we derive, $\mathsf{O}b_1$ ($+\partial_{\mathsf{O}} b_1$), but if in addition we prove that $\neg b_1$ holds, we can conclude that we have a violation, and then we derive the compensatory obligation $\mathsf{O}b_2$, and so on.

The reasoning mechanism is essentially the same as that of basic defeasible presented in Sect. 3.1. The first difference is that an argument can only be attacked by an argument of the same type. Thus if we have an argument consisting of a constitutive rule for p, a counterargument should be a constitutive rule for $\sim p$. The same applies for prescriptive rule. An exception to this is when we have a constitutive rule for p such that all its premises are provable as obligations. In this case the constitutive rule behaves like a prescriptive rule, and can be used as a counterargument for a prescriptive rule for $\sim p$, or the other way around.

Consider, for example, the following two rules

$$r_1 \colon a_1, a_2 \Rightarrow_C b \tag{24}$$

$$r_2 \colon c \Rightarrow_{\mathsf{O}} \neg b. \tag{25}$$

The idea expressed by r_1 is that, in a particular normative system, the combination of a_1 and a_2 is recognised as the institutional fact b, while r_2 prohibits b given c. Suppose now that a_1 and a_2 are both obligatory. Under these conditions it is admissible to assert that b is obligatory as well. Accordingly, r_1 can be used to conclude $\mathsf{O}b$ instead of simply b. This means that the conclusions of r_1 and r_2 are conflicting: thus r_1, when its premises are asserted as obligation, can be used to counter an argument (e.g., r_2) forbidding b (making $\neg b$ obligatory, or $\mathsf{O}\neg b$).

[11] As explained elsewhere [10,22], we do not add a deontic operator in the consequent of rules (i.e., $a_1, \ldots, a_n \Rightarrow \mathsf{O}b$), but we rather differentiate the mode of conclusions by distinguishing diverse rule types. This choice has a technical motivation: (a) it considerably makes simpler and more compact the proof theory; (b) it allows us to characterise a specific logical consequence relation for O, and eventually implement different proof conditions for constitutive rule, and prescriptive rules, also, to account for different burden of proof, and compliance and violations based on the definition of burden presented in [11,12,24] justified by the results by [17].

The second difference is that now the proof tags are labelled with either C, e.g., $+\partial_C p$, (for constitutive conclusions) or with O, e.g., $-\partial_{\mathsf{O}} q$ (for prescriptive conclusions). Accordingly, when we are able to derive $+\partial_{\mathsf{O}} p$ we can say that $\mathsf{O}p$ is (defeasibly) provable.

This feature poses the question of how we model the other deontic operators (i.e., permission and prohibition). As customary in Deontic Logic, we assume the following principles governing the interactions of the deontic operators.[12]

$$\mathsf{O}{\sim}l \equiv \mathsf{F}l \tag{26}$$

$$\mathsf{O}l \wedge \mathsf{O}{\sim}l \rightarrow \bot \tag{27}$$

$$\mathsf{O}l \wedge \mathsf{P}{\sim}l \rightarrow \bot \tag{28}$$

Principle (26) provides the equivalence of a prohibition with a negative obligation (i.e., obligation not). The second and the third are rationality postulates stipulating that it is not possible to have that something and its opposite are at the same time obligatory (27) and that a normative system makes something obligatory and its opposite is permitted (28). (26) gives us the immediate answer on how prohibition is modelled. A rule giving a prohibition can be modelled just as a prescriptive rule for a negated literal. This means that to conclude $\mathsf{F}p$ we have to derive $+\partial_{\mathsf{O}} \neg p$, in other terms that $\neg p$ is (defeasibly) provable as an obligation.

Example 5. Section. 40 of the Australian Road Rules (ARR)[13].

> **Making a U–turn at an intersection with traffic lights**
> A driver must not make a U–turn at an intersection with traffic lights unless there is a U–turn permitted sign at the intersection.

The prohibition of making U-turns at traffic lights can be encoded by the following rule:

$$arr_{40a}\colon \mathit{AtTrafficLigths} \Rightarrow_{\mathsf{O}} \neg \mathit{Uturn}.$$

In a situation where *AtTrafficLights* is given we derive $+\partial_{\mathsf{O}} \neg \mathit{Uturn}$ which corresponds to $\mathsf{F}\,\mathit{Uturn}$.

The pending issue is how to model permissions. Two types of permissions have been discussed in literature following [1,34,35]: (i) weak permission, meaning

[12] In the three formulas below $\rightarrow$ is the material implication of classical logic.

[13] This norm makes use of "must not", to see that "must not" is understood as prohibition in legal documents see, the Australian National Consumer Credit Protection Act 2009, Sect. 29, whose heading is "Prohibition on engaging in credit activities without a licence", recites "(1) A person must not engage in a credit activity if the person does not hold a licence authorising the person to engage in the credit activity".

that there is no obligation to the contrary; and (ii) strong permission, a permission explicitly derogates an obligation to the contrary. In this case we have an exception. For both types of permission we have that the obligation to the contrary does not hold. Defeasible Deontic Logic is capable to handle the two types of permission is a single shot if we establish that $\mathsf{P}p$ is captured by $-\partial_{\mathsf{O}}{\sim}p$. The meaning of $-\partial_{\mathsf{O}}p$ is that p is refuted as obligation, or that it is not possible to prove p as an obligation; hence it means that we cannot establish that p is obligatory, thus there is no obligation contrary to ${\sim}p$.

The final aspect we address is how to model strong permissions. Remember that strong permissions are meant to be exceptions. Exceptions in Defeasible Logic can be easily captured by rules for the opposite plus a superiority relation. Accordingly, this could be modelled by

$$arr_{40e}\colon \mathit{UturnPermittedSign} \Rightarrow_{\mathsf{O}} \mathit{Uturn}.$$

and $arr_{40e} \prec arr_{40a}$. We use a prescriptive defeasible rule for obligation to block the prohibition to U-turn. But, since arr_{49e} prevails over arr_{49a}, we derive that U-turn is obligatory, i.e., $+\partial_{\mathsf{O}}\mathit{Uturn}$.

Thus, when permissions derogate to prohibitions (or obligations), there are good reasons to argue that defeaters for O are suitable to express an idea of strong permission[14]. Explicit rules such as $r\colon a \rightsquigarrow_{\mathsf{O}} q$ state that a is a specific reason for blocking the derivation of $\mathsf{O}{\neg}q$ (but not for proving $\mathsf{O}q$), i.e., this rule does not support any conclusion, but states that ${\neg}q$ is deontically undesirable. Accordingly, we can rewrite the derogating rule as

$$arr_{40e}\colon \mathit{UturnPermittedSign} \rightsquigarrow_{\mathsf{O}} \mathit{Uturn}.$$

In this case, given *UturnPermittedSign* we derive $-\partial_{\mathsf{O}}{\neg}\mathit{Uturn}$.

3.3 Defeasible Deontic Logic Formalised

The presentation in this section is based on [19], and while it is possible to use the standard definitions of *strict rules*, *defeasible rules*, and *defeaters* [2] and the full language presented in [19] for the sake of simplicity, and to better focus on the non-monotonic and deontic aspects, we restrict our attention to defeasible rules and defeaters, and we assume that facts are restricted to plain literals. In addition, we use the equivalences between obligations and prohibitions to transform all prohibitions in the corresponding obligations.

A Defeasible Deontic Theory is a structure

$$(F, R^C, R^{\mathsf{O}}, \prec) \tag{29}$$

where F is a set of facts, where every R^C is a set of constitutive rules, where a constitutive rule obeys the following signature

$$2^{\mathrm{Lit}} \times \mathrm{PLit} \tag{30}$$

[14] The idea of using defeaters to introduce permissions was introduced by [23].

while the signature for R^{O}, a set of prescriptive rules is

$$2^{\mathrm{Lit}} \times \mathrm{Oexpr} \tag{31}$$

Notice that in both cases, the antecedent of a rule can contain both plain and deontic literal, but, in any case, the conclusion is plain literal. Thus, the question is if the conclusions of rules are plain literals, where do we get deontic literals? The answer is that we have two different modes of for the rules. The first mode is that of constitutive rule, where the conclusion is an assertion with the same mode as it appears in the rule (i.e., as an institutional fact); the second mode is that of prescriptive rule, where the conclusion is asserted with a deontic mode (where the deontic mode corresponds to one of the deontic operators).

Constitutive rules behaves as the rules in Basic Defeasible Logic, and we use $\hookrightarrow_C$ to denote the arrow of a constitutive rule. $\hookrightarrow_{\mathsf{O}}$ for the arrow of a prescriptive rule.

For $\otimes$-expression we stipulate that they obey the following properties:

1. $a \otimes (b \otimes c) = (a \otimes b) \otimes c$ (associativity);
2. $\bigotimes_{i=1}^{n} a_i = (\bigotimes_{i=1}^{k-1} a_i) \otimes (\bigotimes_{i=k+1}^{n} a_i)$ where there exists j such that $a_j = a_k$ and $j < k$ (duplication and contraction on the right).

Given an $\otimes$-expression A, the *length* of A is the number of literals in it. Given an $\otimes$-expression $A \otimes b \otimes C$ (where A and C can be empty), the *index* of b is the length of $A \otimes b$. We also say that b appears at index n in $A \otimes b$ if the length of $A \otimes b$ is n.

Given a set of rules R, we use the following abbreviations for specific subsets of rules:

- R_{def} denotes the set of all defeaters in the set R;
- $R[q, n]$ is the set of rules where q appears at index n in the consequent. The set of (defeasible) rules where q appears at any index n is denoted by $R[q]$;
- $R^{\mathsf{O}}[q, n]$ is the set of (defeasible) rules where q appears at index n and the operator preceding it is $\otimes$ for $n > 1$ or the mode of the rule is O for $n = 1$. The set of (defeasible) rules where q appears at any index n is denoted by $R^{\mathsf{O}}[q]$;

A *proof* P in a defeasible theory D is a linear sequence $P(1) \ldots P(n)$ of *tagged literals* in the form of $+\partial_{\Box} q$ and $-\partial_{\Box} q$ with $\Box \in \{C, \mathsf{O}, \mathsf{P}, \mathsf{P_w}, \mathsf{P_s}\}$, where $P(1) \ldots P(n)$ satisfy the proof conditions given in the rest of the paper. The initial part of length i of a proof P is denoted by $P(1...i)$. The tagged literal $+\partial_{\Box} q$ means that q is *defeasibly provable* in D with modality $\Box$, while $-\partial_{\Box} q$ means that q is *defeasibly refuted* with modality $\Box$. More specifically we can establish the following relationships

- $+\partial_C a$: a is provable;
- $-\partial_C a$: a is rejected;
- $+\partial_{\mathsf{O}} a$: $\mathsf{O}a$ is provable;
- $+\partial_{\mathsf{O}} \neg a$: $\mathsf{F}a$ is provable;

- $-\partial_{\mathsf{O}} a$: $\mathsf{O}a$ is rejected;
- $-\partial_{\mathsf{O}} \neg a$: $\mathsf{F}a$ is rejected;
- $+\partial_{\mathsf{P}} a$: $\mathsf{P}a$ is provable;
- $-\partial_{\mathsf{P}} a$: $\mathsf{P}a$ is rejected;
- $+\partial_{\mathsf{P}_\mathsf{s}} a$: $\mathsf{P}_\mathsf{s}a$ is provable;
- $-\partial_{\mathsf{P}_\mathsf{s}} a$: $\mathsf{P}_\mathsf{s}a$ is rejected;
- $+\partial_{\mathsf{P}_\mathsf{w}} a$: $\mathsf{P}_\mathsf{w}a$ is provable;
- $-\partial_{\mathsf{P}_\mathsf{w}} a$: $\mathsf{P}_\mathsf{w}a$ is rejected;

The first thing to do is to define when a rule is applicable or discarded. A rule is *applicable* for a literal q if q occurs in the head of the rule, all non-modal literals in the antecedent are given as facts and all the modal literals have been defeasibly proved (with the appropriate modalities). On the other hand, a rule is *discarded* if at least one of the modal literals in the antecedent has not been proved (or is not a fact in the case of non-modal literals). However, as literal q might not appear as the first element in an $\otimes$-expression in the head of the rule, some additional conditions on the consequent of rules must be satisfied. Defining when a rule is applicable or discarded is essential to characterise the notion of provability for obligations ($\pm\partial_{\mathsf{O}}$) and permissions ($\pm\partial_{\mathsf{P}}$).

A rule $r \in R[q, j]$ is *body-applicable* iff for all $a_i \in A(r)$:

1. if $a_i = \Box l$ then $+\partial_{\Box} l \in P(1...n)$ with $\Box \in \{C, \mathsf{O}, \mathsf{P}, \mathsf{P}_\mathsf{w}, \mathsf{P}_\mathsf{s}\}$;
2. if $a_i = \neg\Box l$ then $-\partial_{\Box} l \in P(1...n)$ with $\Box \in \{C, \mathsf{O}, \mathsf{P}, \mathsf{P}_\mathsf{w}, \mathsf{P}_\mathsf{s}\}$;
3. if $a_i = l \in \text{Lit}$ then $l \in F$.

A rule $r \in R[q, j]$ is *body-discarded* iff $\exists a_i \in A(r)$ such that

1. if $a_i = \Box l$ then $-\partial_{\Box} l \in P(1...n)$ with $\Box \in \{C, \mathsf{O}, \mathsf{P}, \mathsf{P}_\mathsf{w}, \mathsf{P}_\mathsf{s}\}$;
2. if $a_i = \neg\Box l$ then $+\partial_{\Box} l \in P(1...n)$ with $\Box \in \{C, \mathsf{O}, \mathsf{P}, \mathsf{P}_\mathsf{w}, \mathsf{P}_\mathsf{s}\}$;
3. if $a_i = l \in \text{Lit}$ then $l \notin F$.

A rule $r \in R$ is *body-p-applicable* iff

1. if $r \in R^{\mathsf{O}}$ and it is body-applicable; or
2. if $r \in R^C$ and, $A(r) \neq \emptyset$, $A(r) \subseteq \text{PLit}$ and $\forall a_i \in A(r)$, $+\partial_{\mathsf{O}} a_i \in P(1...n)$.

A rule $r \in R$ is *body-p-discarded* iff

1. if $r \in R^{\mathsf{O}}$ and it is not body-applicable; or
2. if $r \in R^C$ and either $A(r) = \emptyset$ or $A(r) \cap \text{DLit} \neq \emptyset$ and $\exists a_i \in A(r)$, $-\partial_{\mathsf{O}} a_i \in P(1...n)$.

The last two conditions are used to determine if we can use a constitutive rule to derive an obligation when all the elements in the antecedent are provable as an obligation but in the constitutive rule require them to be factually derivable, see the discussion for (24) above.

A rule $r \in R[q, j]$ such that $C(r) = c_1 \otimes \cdots \otimes c_n$ is *applicable* for literal q at index j, with $1 \leq j < n$, in the condition for $\pm\partial_{\mathsf{O}}$ iff

1. r is body-p-applicable; and
2. for all $c_k \in C(r)$, $1 \leq k < j$, $+\partial_{\mathsf{O}} c_k \in P(1...n)$ and ($c_k \notin F$ or $\sim c_k \in F$).

Conditions (1) represents the requirements on the antecedent while condition (2) on the head of the rule states that each element c_k prior to q must be derived as an obligation, and a violation of such obligation has occurred.

We are now ready to give the proof conditions for the proof tags used in Defeasible Deontic Logic. The conditions for $+\partial_{\Box}$ and $-\partial_{\Box}$ are related by the Principle of Strong Negation [3,20]: The *strong negation* of a formula is closely related to the function that simplifies a formula by moving all negations to an inner most position in the resulting formula, and replaces the positive tags with respective negative tags, and vice versa. We will give the positive and negative proof conditions for one tag to illustrate how the principle works. For the remaining tag we present only the positive case.

The proof condition of *defeasible provability for a factual conclusion* is

$+\partial_C$: If $P(n+1) = +\partial_C q$ then

(1) $q \in F$ or

(2.1) $\sim q \notin F$ and

(2.2) $\exists r \in R^C[q]$ such that r is body-applicable, and

(2.3) $\forall s \in R[\sim q]$, either

(2.3.1) s is body-discarded, or

(2.3.2) $\exists t \in R^{\mathsf{O}}[q]$ such that t is body-applicable and $s \prec t$.

The proof condition for *defeasible refutability for a factual conclusion* is

$-\partial_C$: If $P(n+1) = -\partial_C q$ then

(1) $q \notin F$ and

(2.1) $\sim q \in F$ or

(2.2) $\forall r \in R^C[q]$ either r is body-discarded, or

(2.3) $\exists s \in R[\sim q]$, such that

(2.3.1) s is body-applicable for q, and

(2.3.2) $\forall t \in R^{\mathsf{O}}[q]$ either t is body-discarded or not $s \prec t$.

The proof conditions for $\pm\partial_C$ are the standard conditions of basic defeasible logic with the proviso that the element of the antecedents of the rules are applicable in the extended deontic logic.

The proof condition of *defeasible provability for obligation* is

$+\partial_{\mathsf{O}}$: If $P(n+1) = +\partial_{\mathsf{O}} q$ then

(1) $\exists r \in R^{\mathsf{O}}_d[q, i] \cup R^C_d[q]$ such that r is applicable for q, and

(2) $\forall s \in R[\sim q, j]$, either

(2.1) s is discarded, or

(2.2) $\exists t \in R[q, k]$ such that t is applicable for q and $s \prec t$.

To show that q is defeasibly provable as an obligation, then q must be derived the prescriptive rules or by rules of the theory that behaves as prescriptive rules in the theory. More specifically, (1) there must be a rule introducing the obligation for q which can apply; this is possible if the rule is a prescriptive rules and all the applicability conditions hold, or there is a constitutive rule where all the conditions of applicability are mandatory (i.e., are provable as obligations).

(2) every rule s for $\sim q$ is either discarded or defeated by a stronger rule for q. If s is an obligation rule, then it can be counterattacked by any type of rule.

The strong negation of the condition above gives the negative proof condition for obligation.

Let us now move to the proof conditions for permission. We start with the condition for strong permission.

The proof condition of *defeasible provability for strong permission* is

$+\partial_{\mathsf{P}_\mathsf{s}}$: If $P(n+1) = +\partial_{\mathsf{P}_\mathsf{s}} q$ then
(1) $+\partial_{\mathsf{O}} q \in P(1...n)$ or
  (2.1) $\exists r \in R^{\mathsf{O}}_{def}[q] \cup R^{C}_{def}[q]$ such that r is body-p-applicable, and
  (2.2) $\forall s \in R[\sim q, j]$, either
    (2.2.1) s is discarded, or
    (2.2.2) $\exists t \in R[q, k]$ such that t is applicable for q and $s \prec t$.

In this case we have two conditions. The first condition is that, we inherit the strong permission from an obligation (see the discussion in Sect. 2). In the second case, the condition has the same structure as that of a defeasible obligation, but instead of a defeasible rule there is a body-applicable prescriptive defeater or body-p-applicable constitutive defeater, that is not defeated (see the discussion in Sect. 3.2, in particular the formalisation of the U-turn at traffic lights example).

The proof condition for weak permission simply boils down to the failure to prove the obligation to the contrary, namely:

$+\partial_{\mathsf{P}_\mathsf{w}}$: If $P(n+1) = +\partial_{\mathsf{P}_\mathsf{w}} q$ then
(1) $-\partial_{\mathsf{O}} \sim q \in P(1...n)$.

Similarly, for a generic permission, a permission that does not distinguish between strong and weak permission, the proof condition is that we have proved either to the two:

$+\partial_{\mathsf{P}}$: If $P(n+1) = +\partial_{\mathsf{P}} q$ then
(1) $+\partial_{\mathsf{P}_\mathsf{s}} q \in P(1...n)$ or
(2) $+\partial_{\mathsf{P}_\mathsf{w}} q \in P(1...n)$.

Notice that in the majority of cases, where a permission appears in the antecedent of a rule the permission is understood as a generic permission.

Finally, we introduce a special proof tag for a distinguished literal meant to represent an non-compensable violation. To this end we extend the set of literal language with the proposition $\bot$. Furthermore, we stipulate that this literal can only appear in the antecedent of rules. As we have seen we can use $\otimes$-expressions to model obligations, their violations, and the corresponding compensatory obligations. Thus, given the $\otimes$-expression

$$a_1 \otimes \cdots \otimes a_n \tag{32}$$

$\mathsf{O}a_1$ is the ideal primary obligations, $\mathsf{O}a_2$ is the second best option when there is a violation of the primary obligation, and if we repeat this argument, a_n is the least one can do to compensate the violations that precede it, and no further compensation is then possible. Thus, we can say that we have obligation

that cannot be compensated if we fully traverse an $\otimes$-expression. Thus, we can say that we have a violation that cannot be compensate if the following proof condition holds.

$+\partial_{\perp}$: If $P(n+1) = +\partial_{\perp}$ then
(1) $\exists R^{\mathsf{O}}{}_d \cup R^C_d$ such that
(1.1) r is body-p-applicable and
(1.2) $\forall c_i \in C(r)$ $+\partial_O c_i \in P(1...n)$ and either $c_i \notin F$ or $\sim c_i \in F$.

We conclude this section by reporting some results about the logic, showing the behaviour of the logic, its deontic and computational properties. In what follows, given a defeasible deontic theory D, we use $D \vdash \pm\#q$, where $\#$ stand for any of the proof tags discussed in this section, to denote that there is a derivation of $\#q$ from D.

Proposition 1. *Given a defeasible deontic theory D.*

1. *it is not possible to have both $D \vdash +\#q$ and $D \vdash -\#q$;*
2. *$D \vdash +\partial_C q$ and $D \vdash +\partial_C \neg q$ if $q, \neg q \in F$;*
3. *for $\Box \in \{\mathsf{O}, \mathsf{P_s}\}$, it is not possible to have both $D \vdash +\partial_{\Box} q$ and $D \vdash +\partial_{\Box} \sim q$;*
4. *for $\Box \in \{\mathsf{O}, \mathsf{P_s}\}$, if $D \vdash +\partial_{\Box} q$, then $D \vdash -\partial_{\Box} \sim q$;*
5. *it is not possible to have both $D \vdash +\partial_{\mathsf{O}} q$ and $D \vdash +\partial_{\mathsf{P_s}} \sim q$;*
6. *for $\Box \in \{\mathsf{P}, \mathsf{P_s}, \mathsf{P_w}\}$, if $D \vdash +\partial_{\mathsf{O}} q$, then $D \vdash +\Box q$;*
7. *if $D \vdash +\partial_{\mathsf{P_w}} q$, then $D \vdash -\partial_{\mathsf{O}} \sim q$.*

The first property is an immediate consequence of using the Principle of Strong Negation in formulating the proof conditions. Properties 2 and 3 establish the consistency of the logic. The inference mechanisms cannot produce an inconsistency unless the monotonic part (facts) is already inconsistent. Property 4 is again a corollary of the Principle of Strong Negation. Finally, Properties 5 and 6 show that the deontic operators satisfy the properties normally ascribed to them in deontic logics.

Given a Defeasible Theory D, HB_D is the set of literals such that the literal or its complement appears in D, where 'appears' means that it is a sub-formula of a modal literal occurring in the theory. The modal Herbrand Base of D is $HB = \{\Box l |\ \Box \in \{C, \mathsf{O}, \mathsf{P}, \mathsf{P_w}, \mathsf{P_s}\}, l \in HB_D\}$. Accordingly, the extension of a Defeasible Theory is defined as follows.

Given a Defeasible Theory D, the *defeasible extension* of D is defined as

$$E(D) = (+\partial_{\Box}, -\partial_{\Box})$$

where $\pm\partial_{\Box} = \{l \in HB_D : D \vdash \pm\partial_{\Box} l\}$ with $\Box \in \{C, \mathsf{O}, \mathsf{P}, \mathsf{P_w}, \mathsf{P_s}\}$. In other words, the extension is the (finite) set of conclusions, in the Herbrand Base, that can be derived from the theory.

Proposition 2. *Given a defeasible theory D, the extension of the theory $E(D)$ can be computed in $O(size(D))$. Where the size of the theory D, $size(D)$ is determined by the number of occurrences of literals in the theory.*

The procedure to compute the extension in polynomial (linear) time is given in [19] and it is based on the algorithm given by Maher [28] to show that computing the extension of a basic defeasible theory can be computed in linear time.

4 Modelling Norms with Defeasible Deontic Logic

We begin this paper with a scenario, Example 1, that has been used to identify limitations of other approach to model real life norms (to be precise, it is an extension of the example to show the drawback of other approach to model norms using formalism based on possible world semantics, see [14,15]). Let us see how to model it in Defeasible Deontic Logic.

The first step is to understood the intent of the contract. We can paraphrase the contract as follows:

C0: the use of the product is forbidden;
C1: if a license is granted, then the use of the product is permitted;
C2: the publication of the result of the evaluation is forbidden;
C2c: the removal of the illegally published results within the allotted times compensates the illegal publication;
C2e: the publication of the results of the evaluation is permitted if approval is obtained before publication;
C3: commenting about the evaluation is forbidden;
C3e: commenting about the evaluation is permitted, it publication of the results is permitted;
C4: publication of the results of evaluation is obligatory, if commissioned for an independent evaluation;
C4x: the use of product is obligatory, if commissioned for an independent evaluation;
C5: the use of the product is forbidden, if there is a violation of the above conditions.

Based on the decomposition above we can formulate the following rules and instances of the superiority relation:

$$
\begin{aligned}
r_0 &: \Rightarrow_{\mathsf{O}} \neg use \\
r_1 &: license \rightsquigarrow_{\mathsf{O}} use \\
r_2 &: \Rightarrow_{\mathsf{O}} \neg publish \otimes remove \\
r_{2e} &: approval \rightsquigarrow_{\mathsf{O}} publish \\
r_3 &: \Rightarrow_{\mathsf{O}} \neg comment \\
r_{3e} &: \mathsf{P} publish \rightsquigarrow_{\mathsf{O}} comment \\
r_4 &: commission \Rightarrow_{\mathsf{O}} publish \\
r_{4x} &: commission \Rightarrow_{\mathsf{O}} use \\
r_5 &: \bot \Rightarrow_{\mathsf{O}} \neg use
\end{aligned}
$$

where $r_O \prec r_1$, $r_o \prec r_{4x}$, $r_1 \prec r_5$, $r_{4x} \prec r_5$ $r_2 \prec r_{2e}$, $r_3 \prec r_{3e}$.

The situation problematic for other approaches is the case, that the licensee, with a valid license, illegally publishes the results of the evaluation, remove them, and at the same time post a comment about the evaluation. The other approaches conclude that commenting is permitted, but the permission to post

a comment requires the permission to publish, but since the publishing was illegal, there was not permission to publish. For us, if we assume, the facts *license*, *publish*, *remove* and *comment*, then from r_1 we derive $\mathsf{P}use$, from r_2, we have $\mathsf{F}publish$, and $\mathsf{O}remove$. Given that we have $\mathsf{F}publish$ we cannot conclude $\mathsf{P}publish$, thus, $-\partial_{\mathsf{P}}publish$ hence from r_3, we obtain $\mathsf{F}comment$, and thus the conditions to derive $+\partial_{\perp}$ indicating that the situation is not compliant.

5 Conclusion and Summary

We began this paper with some simple and small examples containing essential features for normative reasoning. We discussed, why defeasible reasoning, supplemented with deontic operators has the capability to offer a conceptually sound and computationally feasible tool for legal reasoning (or aspects of it). Then, we proposed a specific logic, Defeasible Deontic Logic, and we show how to capture the various features in the logic, we illustrated some of the feature with legal examples. The logic presented in the paper has a full implementation [27], and it has been used in a number of applications and industry scale pilot projects (for example to determine the regulatory compliance of business processes [13]). Recently, the logic has been proposed for the reasoning engine for a platform to handle legislations in a digital format in the Regulation as a Platform (RaaP) project funded by the Australian Federal Government. As part of the project, several Acts have been encoded in Defeasible Deontic Logic demonstrating the suitability of the logic to model normative systems. The Regulation as a Platform framework is available for public evaluation at https://raap.d61.io.

References

1. Alchourrón, C.E., Bulygin, E.: Permission and permissive norms. In: Krawietz, W., et al. (ed.) Theorie der Normen. Duncker & Humblot (1984)
2. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: Representation results for defeasible logic. ACM Trans. Comput. Log. **2**(2), 255–287 (2001)
3. Antoniou, G., Billington, D., Governatori, G., Maher, M.J., Rock, A.: A family of defeasible reasoning logics and its implementation. In: Horn, W. (ed.) In: Proceedings of the 14th European Conference on Artificial Intelligence, ECAI 2000, Amsterdam, pp. 459–463. IOS Press (2000)
4. Billington, D., Antoniou, G., Governatori, G., Maher, M.J.: An inclusion theorem for defeasible logic. ACM Trans. Comput. Log. **12**(1), 6 (2010)
5. Carmo, J., Jones, A.J.I.: Deontic logic and contrary-to-duties. In: Gabbay, D.M., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 8, pp. 265–343. Springer, Dordrecht (2002). https://doi.org/10.1007/978-94-010-0387-2_4
6. Chisholm, R.M.: Contrary-to-duty imperatives and deontic logic. Analysis **24**, 33–36 (1963)
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2), 321–358 (1995)

8. Gordon, T.F., Governatori, G., Rotolo, A.: Rules and norms: requirements for rule interchange languages in the legal domain. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 282–296. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04985-9_26
9. Gordon, T.F., Prakken, H., Walton, D.: The Carneades model of argument and burden of proof. Artif. Intell. **171**(10–11), 875–896 (2007)
10. Governatori, G.: Representing business contracts in RuleML. Int. J. Coop. Inf. Syst. **14**(2–3), 181–216 (2005)
11. Governatori, G.: On the relationship between Carneades and defeasible logic, pp. 31–40. ACM (2011)
12. Governatori, G.: Burden of compliance and burden of violations. In: Rotolo, A. (ed.) 28th Annual Conference on Legal Knowledge and Information Systems, Frontieres in Artificial Intelligence and Applications, Amsterdam, pp. 31–40. IOS Press (2015)
13. Governatori, G.: The Regorous approach to process compliance. In: 2015 IEEE 19th International Enterprise Distributed Object Computing Workshop, pp. 33–40. IEEE Press (2015)
14. Governatori, G.: Thou shalt is not you will. In: Atkinson, K. (ed.) Proceedings of the Fifteenth International Conference on Artificial Intelligence and Law, pp. 63–68. ACM, New York (2015)
15. Governatori, G., Hashmi, M.: No time for compliance. In: Hallé, S., Mayer, W. (ed.) 2015 IEEE 19th Enterprise Distributed Object Computing Conference, pp. 9–18. IEEE (2015)
16. Governatori, G., Idelberg, F., Milosevic, Z., Riveret, R., Sartor, G., Xu, X.: On legal contracts, imperative and declarative smartcontracts, and blockchain systems. Artif. Intell. Law 1–33 (2018)
17. Governatori, G., Maher, M.J.: Annotated defeasible logic. Theory Pract. Log. Program. **17**(5–6), 819–836 (2017)
18. Governatori, G., Maher, M.J., Billington, D., Antoniou, G.: Argumentation semantics for defeasible logics. J. Log. Comput. **14**(5), 675–702 (2004)
19. Governatori, G., Olivieri, F., Rotolo, A., Scannapieco, S.: Computing strong and weak permissions in defeasible logic. J. Philos. Log. **42**(6), 799–829 (2013)
20. Governatori, G., Padmanabhan, V., Rotolo, A., Sattar, A.: A defeasible logic for modelling policy-based intentions and motivational attitudes. Log. J. IGPL **17**(3), 227–265 (2009)
21. Governatori, G., Rotolo, A.: Logic of violations: a Gentzen system for reasoning with contrary-to-duty obligations. Australas. J. Log. **4**, 193–215 (2006)
22. Governatori, G., Rotolo, A.: Changing legal systems: legal abrogations and annulments in defeasible logic. Log. J. IGPL **18**(1), 157–194 (2010)
23. Governatori, G., Rotolo, A., Sartor, G.: Temporalised normative positions in defeasible logic. In: Gardner, A. (ed.) 10th International Conference on Artificial Intelligence and Law, pp. 25–34. ACM Press (2005)
24. Governatori, G., Sartor, G.: Burdens of proof in monological argumentation. In: Winkels, R. (ed.) The Twenty-Third Annual Conference on Legal Knowledge and Information Systems, Volume 223 of Frontiers in Artificial Intelligence and Applications, Amsterdam, pp. 57–66. IOS Press (2010)
25. Grossi, D., Jones, A.: Constitutive norms and counts-as conditionals. Handb. Deontic Log. Norm. Syst. **1**, 407–441 (2013)

26. Idelberger, F., Governatori, G., Riveret, R., Sartor, G.: Evaluation of logic-based smart contracts for blockchain systems. In: Alferes, J.J., Bertossi, L., Governatori, G., Fodor, P., Roman, D. (eds.) RuleML 2016. LNCS, vol. 9718, pp. 167–183. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42019-6_11
27. Lam, H.-P., Governatori, G.: The making of SPINdle. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 315–322. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04985-9_29
28. Maher, M.J.: Propositional defeasible logic has linear complexity. Theory Pract. Log. Program. **1**(6), 691–711 (2001)
29. Nute, D.: Defeasible logic. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 3, pp. 353–395. Oxford University Press, Oxford (1994)
30. Palmirani, M., Governatori, G., Athan, T., Boley, H., Paschke, A., Wyner, A. (eds.) LegalRuleML Core Specification Version 1.0. OASIS (2017)
31. Prakken, H., Sergot, M.: Contrary-to-duty obligations. Studia Logica **57**(1), 91–115 (1996)
32. Sartor, G.: Legal Reasoning: A Cognitive Approach to the Law. Springer (2005)
33. Searle, J.R.: The Construction of Social Reality. The Free Press, New York (1996)
34. Soeteman, A.: Logic in Law. Kluwer, Dordrecht (1989)
35. von Wright, G.H.: Norm and Action: A Logical Inquiry. Routledge and Kegan Paul, Abingdon (1963)