Deontic Meta-Rules

Francesco Olivieri[1], Guido Governatori[2], Matteo Cristani[3], Antonino Rotolo[4], and Abdul Sattar[1]

[1] School of Information and Communication Technology, IIIS, Griffith University, Nathan, QLD 4111, Australia {f.oliveri;a.sattar}@griffith.edu.au
[2] guido@governatori.net
[3] University of Verona, Verona, 37136, Italy matteo.cristani@univr.it
[4] Alma AI, University of Bologna, Bologna, 40121, Italy antonino.rotolo@unibo.it

Abstract. The use of meta-rules in logics, i.e., rules whose content includes other rules, has recently gain attention in the setting of non-monotonic reasoning: the authors of [66] proposed a first logical formalisation, by also advancing efficient algorithms that compute the (meta)-extensions of such theories. The aim of this work is to extend such a logical framework by considering the deontic aspect. The resulting logic will not just be able to model policies, but also to tackle well-known aspects that occur in numerous legal systems, such as the notion of *normative power*, that is the ability of an agent to create new norms affecting other agents. The use of Defeasible Logic (DL) to model meta-rules in the application area we just alluded to has been investigated. Within this line of research, the study mentioned above was not focusing on the general computational properties of meta-rules.
This study undertakes an ample effort to fill this gap as we shall advance three major contributions. First, we introduce and formalise two variants of Defeasible Deontic Logic with Meta-Rules to represent (1) defeasible meta-theories with deontic modalities, and (2) two different types of conflicts among rules: Simple Conflict Defeasible Deontic Logic, and Prudential Conflict Defeasible Deontic Logic. We advance efficient algorithms to compute the extensions for both variants. Finally, we prove consistency and coherency of the logics, and correctness, completeness, and computational complexity of such algorithms.

1 Introduction and Background

An extensive research has been devoted in the AI community for developing rule-based systems in the normative domain. It is widely acknowledged that norms have, basically, a conditional structure like

$$\textsc{If } a_1, \ldots, a_n \textsc{ Then } b$$

where $a_1, \ldots, a_n$ are the applicability conditions of the norm, and b denotes the legal effect which ought to follow when such applicability conditions hold. This, very general, view highlights an immediate link between the concepts of norm and rule.

An important aspect of normative reasoning is the introduction of meta-norms (i.e., meta-rules): Norms conferring the power to produce other norms, justifying their choice or their enactment, as well as governing their application and dynamics [44,70,27].

This paper develops, in the context of normative reasoning, a rule-based logical framework providing an efficient computation of meta-rules, i.e., rules having rules as their elements. We shall proceed with the simple idea that a rule is a (binary) relation between a set of (applicability) conditions and a conclusion. The meaning of such a relation is to determine under which conditions a conclusion can be derived. Meta-rules thus generalise such an idea by establishing that, in addition to standard conclusions and conditions, rules themselves can be the "conclusion" and the elements of the set of conditions. In other terms, new rules can be generated from other rules, along with conditions that are not rules per se.

An interesting discussion can be developed in the normative domain, as meta-rules (or rules with nested rules) frequently occur not only in normative reasoning, but in other contexts as well including, for instance, policies for security systems. Often, when a set of policies is represented by a set of rules, we have to consider a policy that contains conditions (rules) about *itself* (and also about other sets of rules/policies). Consider the example provided in [75], where a company has a security policy specifying that: (i) a piece of information is deemed confidential when its disclosure would harm the interests of the company, and (ii) confidential information must be protected (and hence cannot be disclosed). Such a policy can be naturally represented by the meta-rule

$$\big(\mathit{Disclose}(x) \to \mathit{HarmInterests}\big) \to \mathit{Confidential}(x).$$

Now, in this policy, the condition about harming the interests should be represented by an hypothetical expression: an 'If (...) Then (...)' rule would appear to be the most natural way to represent such a construct. Furthermore, the hypothesis is part of the conditions to define when a piece of information is confidential (in this case, is the condition itself). Unfortunately, we cannot use classical material implication ($\supset$), given its well-known paradoxes. Consequently, if we model the policy as

$$(\mathit{Disclose}(x) \supset \mathit{HarmInterests}) \to \mathit{Confidential}(x),$$

given the equivalence between '$\mathit{Disclose}(x) \supset \mathit{HarmInterests}$' and '$\neg\mathit{Disclose}(x) \vee \mathit{HarmInterests}$', we have the counter-intuitive scenarios where (1) if x is not disclosed then x is confidential (information that is not confidential, no matter if disclosed or not, it does not need to be protected), and (2) if, for any reason, company interests are harmed, then x is confidential, for any piece of information. The policy can neither be defined as

$$\big(\mathit{Disclose}(x) \wedge \mathit{HarmInterests}\big) \to \mathit{Confidential}(x),$$

given that a disclosed information (with the consequent harm of interests) can no longer be considered confidential.

Another situation where meta-rules are useful is when the inclusion of a rule in a set of rules depends upon whether other rules are already in the system.

For instance, we can have

$$r_1 \rightarrow r_2,$$

indicating that the existence of rule r_2 in the system depends upon the existence in the system of rule r_1. For instance, this could be the case of an *implementing decree* whose mere existence depends upon the existence of the law whose prescriptions are indeed implemented.

Typically, such dependencies among rules are stored externally, but if we model them directly into the system by using meta-rules, then we can include (or remove) r_2 automatically, based on the other rules it depends upon (and thus automatising the system's maintenance functionalities). In addition, this feature is beneficial to system integration, as it supports context-dependent rules.

The definition of context-dependent policies is valuable in many situations; for instance, the defence forces of a country can have different rules of engagement, depending on the environment in which they are situated. One might think that a simple (and somehow naive) way to achieve this would be in partitioning the rules into independent rule sets, one for each context, and then to use simpler rules (without nested rules). However, as discussed before, there could be dependencies among the rules, and the environments themselves could be defined in terms of rules. Accordingly, a clear partition of the simple rules might not be feasible.

Another area of legal reasoning, where meta-rules proved to be essential in representing the legal processes at hand, is related to the of field of norm change. As we argued (see also [41]), many legislative instruments contain provisions (norms) about who has the power to create, modify, revoke, or abrogate other norms. If norms can be represented as rules [73], and there are norms 'about other norms', then it would be natural to have rules whose content consists of other rules. Different variants of Defeasible Logic have been proposed [41,17] to incorporate meta-rules in order to describe the logical behaviour of norm changes. An important aspect of norm change is that some legal systems can specify that certain norms cannot exist (or cannot be in force) in a specific legal system. For instance, in the Italian Constitution, Article 27 prescribes that there cannot be norms in the Italian legal system prescribing Capital Punishment. This means that a meta-norm can speak about the positive existence of a rule, as well as preventing a rule to be generated in the system itself after the enforcement of the meta-norm.

Finally, meta-rules have been proposed to model private international law, which is the body of rules and principles governing the choice of law to be applied, when there are conflicts in the domestic law of different countries related to private legal facts and transactions [77]. Works such as [20,55,56] argued that we need a reasoning mechanism that allows us

- To conclude that if something holds in some legal system, then some norms hold in this or another legal system;
- To import, in a given system, any piece of information holding in another system.

Meta-rules are thus relevant in order to model such an interaction.

In this paper, we extensively study a new computational framework (based on a variant of Defeasible Deontic Logic) that models rules and meta-rules, for the normative domain.

We shall distinguish between the *content* of a rule (which is a binary relation between a set of premises and a conclusion, both represented as propositions in an underlying, given language), and the *name* (or *label*, the identifier) of the rule itself. In this set up, a rule can be understood as a function associating a label to the content of the rule. Similarly, a meta-rule is still a function that associates a label to the content, but now the elements of the binary relation corresponding to the content of the meta-rule can contain other rules.

2 Synopsis and Structure of the Paper

In devising a new computational logical system for meta-rules in deontic reasoning, we shall adopt the following road map and take the following steps:

- We offer a conceptual analysis of meta-reasoning in the normative domain;
- We present the new logical framework;
- We investigate the computational properties of the logic.

We thus commence with Section 3 by setting up a conceptual framework, where some basic philosophical problems are discussed; such a conceptual framework provides a discussion that frames meta-rules in legal reasoning, and is hence the basis for our formal choices. For the sake of illustration, we discuss and consider (1) what may be the complexities behind the definition of norm change mechanisms in domains such as the law, and (2) the interplay of norm dynamics and deontic concepts such as permissions and permissive norms.

Once this is done, we move on in providing the proper formalisation of the logical apparatus that models deontic and normative reasoning, and that which is enriched with meta-rules. Such a formalism must move from the above-mentioned conceptual analysis, and must be as richest as possible.

To help the reader, the logic is presented progressively. As the technicalities of the proof theoretic part can be harsh to grasp for a non-expert reader, a gentle introduction to the formal machinery is thus offered: in particular, we split the logic presentation and formalisation in two distinct sections.

Section 4 presents the Defeasible Deontic Logic's framework of [35], and its main purpose is twofold. First, it introduces the reader to all the notions (some specific of Defeasible Logic) that will be needed later one, such as: (i) explaining the meaning of a rule being applicable or discarded, and their meaning/importance, (ii) representing the different types of rules - strict, defeasible and defeaters - (iii) modelling the distinction between constitutive and deontic rules, (iii) how such a framework is capable to solve conflicts, or again (iv) handling complex reasoning patterns related to contrary-to-duty and compensatory obligations (as well as various types of deontic statements). Second, all those notions are formalised: language, strict and defeasible rules vs constitutive, obligation and permission rules, and so on. For every definition and concept, we thoroughly explain their meaning and propose examples.

Follows Section 5 where we present Defeasible Deontic Logic with Meta-Rules: such a framework specialises the framework of [35] by (i) representing the distinction between rules and meta-rules, (ii) introducing and formalising different *variants* of *conflict* among meta-rules, and naturally (iii) providing the proof theory extended with meta-rules and such conflicts. We end that section by proving coherence and consistence of the logical apparatus.

Lastly, we prove that the logic proposed is correct and computationally efficient. Section 6 thus advances algorithms that compute the extension of a given input logic, and explain their behaviours through two examples. We end that section by studying their complexity and proving their correctness.

This work is ended by Section 7, that reviews current relevant literature, and by Section 8 where we discuss how this research can be taken further.

3 The Conceptual Framework

It was argued in [41] that meta-rules can describe the dynamics of any normative institution (such as a legal system) where norms are formalised and can be used to establish conditions for the creation and modification of other rules or norms. In turn, proper rules precisely correspond to norms in normative systems. In particular, it was pointed out that meta-rules can be represented in the language of Defeasible Logic as follows:[5]

$$m\alpha\colon a_1, \ldots, a_n \Rightarrow (\beta\colon b_1, \ldots, b_m \Rightarrow c),$$

precisely to grasp norm change mechanisms in the law. For instance, if the rule '$\beta\colon b_1, \ldots, b_m \Rightarrow c$' does not exist in the theory at hand, then the successful application of $m\alpha$ leads to derive such a rule, which amounts to enact β as a new norm in the legal system. Similarly, if β already exists but has the form '$\beta\colon b_1 \Rightarrow c$', then the successful application of $m\alpha$ corresponds to modifying β from '$\beta\colon b_1 \Rightarrow c$' into '$\beta\colon b_1, \ldots, b_m \Rightarrow c$'.

In addition, with meta-rules we can admit the *negation of rules*. If we are able to conclude that a (positive) rule holds, then it means that we can insert the rule (the content of the rule, with a specific name) in the system, and we can use the resulting rule to derive new conclusions. For a negated rule, the meaning is that it is not possible to obtain a rule with that specific content (this can be either formally prescribed for the whole rule, or in a way that results irrespective of the name).

In this paper we go further, and discuss the conceptual and logical meanings of modalising rules via meta-rules, i.e., what we mean when we establish the obligatoriness of the enactment of a certain rule. Put it in this way, making a rule obligatory amounts to firing a meta-rules like the following:

$$m\zeta\colon a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} (\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c).$$

[5] From now on, as formalised in the language in the two following sections, literals will be denoted by Latin letters, standard rules by Greek letters, and meta-rules by Greek letters preceded by lower-case 'm'.

Rule γ is a standard deontic rule as described in previous works [37,35,40]: if $a_1, \ldots, a_n$ are the case, then rule γ allows us to derive that c is obligatory. If a similar intuition is extended to meta-rules, then $m\zeta$ asserts that enacting rule '$\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c$' is obligatory.

We should now ask: For whom '$\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c$' is obligatory? There is no unique answer, since the question has different meanings, depending on which normative domain is considered. If we work, again, in the legal domain, we can imagine that a legislative authority imposes the obligation to enact a certain norm over another authority, lower in the legislative hierarchy but competent in enacting such a norm. Concrete examples occur in the law, when, for instance, European authorities impose to member states the implementation of European directives, or when Constitutional courts require national parliaments to amend, or to integrate, the legislative corpus.

In this investigation, we aim to unravel some important aspects of meta-norms, looking closely at the *mechanism* that allows meta-rules to develop rules, or to prevent their existence. In the current investigation, we are not concerned with the complex aspects underlying the *goals* (and potentially the *intentions*) of these meta-rules. This is a different issue, which has been partly explored in the recent literature [17] but it is not the focus of this study.

We remark that structures like $m\zeta$ do not lead to a meaningless iteration of deontic modalities, an issue that was explored in early deontic literature such as in [6,7]. First of all, the obligatory enactment of a prescriptive norm is not equivalent to a simple iteration of obligations. In addition, while it looks meaningless to have an expression like OOc standing for "It is obligatory that it is obligatory that everyone keeps their promises", an expression like "Parking on highways ought to be forbidden" makes sense [6]. The latter example suggests that a norm forbidding to park in highways is obligatory, thus assuming a conceptual distinction between norms, on the one side, and obligations (and permissions) on the other side [2,52]. Our general approach, hence, clearly distinguishes norms from obligations: obligations are the effects (i.e., the conclusions) of the application of prescriptive norms. Under this reading, the application of rule γ leads to the obligation Oc, while the application of meta-rule $m\zeta$ leads to state where norm γ ought to be the case.

What about the permission? The elusive character of permission affects also the case of permissive meta-rules. Consider the meta-rule

$$m\xi\colon a_1, \ldots, a_n \Rightarrow_{\mathrm{P}} (\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c).$$

Does the well-known distinction between *weak* and *strong* permission [79] apply to this case as well? In the standard scenario, the former type of permission corresponds to saying that something is allowed by a code precisely when it is not prohibited by that code. This idea is preserved when meta-rules are considered. Indeed, one may simply argue that, if the legal system *does not* support the derivation of neither

$$\mathrm{O}\neg(\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c)$$
$$\mathrm{O}(\varphi\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} \neg c),$$

then one may also conclude that the following holds:

$$\mathrm{P}(\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c).$$

For similar reasons, since weak permission is the dual of obligation – i.e., $\mathrm{O}A =_{def} \neg\mathrm{O}\neg A$ – imposing consistency means $\mathrm{O}A \rightarrow \mathrm{P}A$, and it hence looks reasonable that from '$\mathrm{O}(\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c)$' we can obtain '$\mathrm{P}(\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c)$'.

As is well-known, the concept of strong permission is more complicated, as it amounts to saying that some a is permitted by a code iff such that code explicitly states that a is permitted. Various sub-types of permissions can be thus identified, such as the following ones [53,10,76,35]:

Static permission: X is a static permission, wrt a normative system, when X is derived from a strong permission, i.e., from an explicit permissive norm.

Dynamic permission: X is a dynamic permission, wrt a normative system, when it guides the legislator by describing the limits on what may be prohibited typically without violating static permissions in the system.

Exemption: X is an exemption, wrt a normative system, when it is an exception of a prohibition contained in the system.

It seems reasonable that all these types may be extended to the case of meta-rules. The first case simply amounts to when a rule is explicitly permitted in the theory, or it is derived from a permissive meta-rule (such as mr_3). The third case is when, for instance, we have in the system two rules such as

$$\mathrm{P}(\gamma : b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c)$$
$$\mathrm{O}(\varphi : b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} \neg c),$$

and we know that γ is stronger than φ, or, at a meta-rule level, we have

$$m\xi\colon a_1, \ldots, a_n \Rightarrow_{\mathrm{P}} (\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c)$$
$$m\chi\colon a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} (\varphi\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} \neg c)$$

and, similarly, $m\xi$ is stronger than $m\chi$. The idea of dynamic permission consists in preventing the theory from deriving any incompatible deontic rule, for example, by setting that the meta-rule $m\xi$ is stronger than any other meta-rule supporting any conflicting rule.

The discussion above shows how crucial it is to establish some logical properties of permitted rules and to determine when modalised rules are in conflict. Accordingly, desirable basic properties are the following ones:

$$\mathrm{P}(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b) \vdash \neg\mathrm{O}(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} \neg b) \quad (1)$$
$$\mathrm{P}(a_1, \ldots, a_n \Rightarrow_{\Box} b) \vdash \neg\mathrm{O}\neg(a_1, \ldots, a_n \Rightarrow_{\Box} b), \quad (2)$$

where (2) is trivially desirable because, for a rule ϕ, $\mathrm{P}\phi$ should imply $\neg\mathrm{O}\neg\phi$. Instead, if the rule in the scope of the permission is a permissive norm (i.e., '$a_1, \ldots, a_n \Rightarrow_{\mathrm{P}} b$'), then principle (1) does not hold in general[6]. Suppose that the normative system (for instance, on the basis of constitutional values) permits the enactment of a norm that allows for the temporary limitation of liberties due

[6] Although we may have concrete examples where it seems a bit odd that the legislator explicitly issues *both* '$a_1, \ldots, a_n \Rightarrow_{\mathrm{P}} b$' and '$a_1, \ldots, a_n \Rightarrow_{\mathrm{P}} \neg b$', this is deontically possible: see below. Many thanks to a reviewer for commenting on this point.

to public health reasons. If so, it would not be contradictory to have that the normative system prescribes that the lack of limitation is also permitted. The following rules are in fact not necessarily incompatible:

$$\textit{public\_health} \Rightarrow_{\mathrm{P}} \textit{limit\_liberties}$$
$$\textit{public\_health} \Rightarrow_{\mathrm{P}} \neg\textit{limit\_liberties}$$

which, essentially, amount to saying that, under certain conditions, P*limit_liberties* and P¬*limit_liberties* are deontically compatible.

Other properties depend upon at to what extent we assume that the legislator was rational in a subtler way. For instance, the following one

$$\mathrm{P}(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b) \vdash \neg\mathrm{P}(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} \neg b) \tag{3}$$

can be rational, since it does not make much sense for a rational legislator to permit that two deontically incompatible rules are the case.

Let us discuss further this case. If we accept it, one may similarly argue that we should also adopt the following (where b is a literal):

$$\mathrm{P}b \vdash \neg\mathrm{P}\neg b \tag{4}$$

which, however, cannot be accepted. Why? Because the deontic facultativeness of a certain b precisely amounts to stating $\mathrm{P}b \wedge \mathrm{P}\neg b$, so (4) would make facultativeness inconsistent.

What about facultative norms? We may have two cases like:

$$\mathrm{P}(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b) \qquad \mathrm{P}\neg(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b) \tag{5}$$
$$\mathrm{P}(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b) \qquad \mathrm{P}(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} \neg b). \tag{6}$$

Clearly, (5) and (6) are deontically different. While the former states that '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b$' is facultative, the latter states that two deontically incompatible and different rules are permitted.

What is facultative according to (6)? Certainly we cannot say that Ob is facultative, since we would need to permit two rules, one supporting Ob – which is the case: '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b$' – and one supporting ¬Ob – which we do not have because '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} \neg b$' rather supports O¬$b$. Principle (6) licenses that both Ob and O¬b are permitted given '$a_1, \ldots, a_n$'. Is it factually possible? Yes, it is. Is it deontically admissible? Perhaps, it is. Is it always deontically rational? We have sometimes arguments to answer: No.

Consider the idea of static permission. Assume we would like to avoid having two deontically incompatible rules in the system. Then, it would not be reasonable to explicitly permit that both rules are the case. As in sceptical defeasible reasoning, if we have arguments for deriving b and ¬b, we refrain from concluding anything on the assumption that b and ¬b are in contradiction; we could reason similarly when two conflicting deontic rules like '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b$' and '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} \neg b$' are taken into account.[7]

[7] One may argue that, if the conflict between '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b$' and '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} \neg b$' cannot be solved, this would imply (but would not be equivalent in Defeasible Logic to) b being weakly permitted. Therefore, we would make a weird use of static positive permissions to state that b is weakly permitted.

Consider instead the idea of dynamic permission: the aim here is limiting the legislator in dynamically adding prohibitions. If so, we would have that

$$\mathrm{P}(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b) \text{ prevents the derivation of } a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} \neg b$$

$$\mathrm{P}(a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} \neg b) \text{ prevents the derivation of } a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b.$$

If we have it (i.e., we prevent both derivations), it means that we want the system to be deontically indifferent with respect to b whenever '$a_1, \ldots, a_n$' are the case. Suppose however that we also have other two norms like the following:

$$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} c$$

$$\mathrm{O}c \Rightarrow_{\mathrm{O}} b.$$

Despite the fact that we prevent the derivation of both '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} \neg b$' and '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b$', can we still say that the system is deontically indifferent with respect to b given '$a_1, \ldots, a_n$'?

Accordingly, under the reading above, one may prudentially establish that the following meta-rules are somehow incompatible:

$$m\xi\colon a_1, \ldots, a_n \Rightarrow_{\mathrm{P}} (\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c) \tag{7}$$

$$m\chi\colon a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} (\varphi\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} \neg c) \tag{8}$$

In the above discussion, we assume that P is not the dual of O, but rather as another $\Box$-operator. A logic for meta-rules assuming that 7 and 8 *are in conflict* is named **prudential**.

On the contrary, if we believe that the permission, as applied to rules, behaves exactly as when literals are modalised, then $m\alpha$ and $m\zeta$ are not in conflict, and this holds precisely because

$$\xi\colon a_1, \ldots, a_n \Rightarrow_{\mathrm{P}} b \tag{9}$$

$$\psi\colon a_1, \ldots, a_n \Rightarrow_{\mathrm{P}} \neg b \tag{10}$$

are likewise compatible in virtue of the intuition that $\mathrm{P}b$ and $\mathrm{P}\neg b$ are consistent. A logic for meta-rules assuming that 7 and 8 *are not in conflict* is named **simple**.

Finally, let us comment on the application of the $\otimes$ operator [35] to rules. In the standard reading of this operator, a rule like '$a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} b \otimes c$' means that if '$a_1, \ldots, a_n$' are the case, then b is obligatory; on the contrary, if the obligation b is not fulfilled, then the obligation c is activated, and becomes in force until it is satisfied or violated. Since we argued that a legislator L_1 can impose to another legislator L_2 to enact a norm μ, we can imagine a scenario where L_2 violates $\mathrm{O}\mu$, but we can also imagine that L_1 has considered a sanction as the result of such a violation, or rather that another normative solution is advanced by L_2. Accordingly, expressions such as

$$m\alpha\colon a_1, \ldots, a_n \Rightarrow_{\mathrm{O}} (\gamma\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} c) \otimes (\zeta\colon b_1, \ldots, b_m \Rightarrow_{\mathrm{O}} d) \tag{11}$$

are admissible. The $\otimes$ operator can be also seen as a preference operator, where the first element is the most preferred, and the last is the least of the acceptable options [35,37]. According to this reading, meta-rule $m\alpha$ establishes that the underlying normative systems should introduce rule β (if such a rule is not already in the system). Alternatively, a less preferable, but still acceptable, outcome is to impose ζ. Generally, in real-life normative systems, the idea is to use this kind of structure to prescribe more and more stringent norms/policies.

4 Defeasible Deontic Logic

Standard Defeasible Logic (SDL) [61,3] is a simple, flexible, and efficient rule-based non-monotonic formalism. Its strength lies in its constructive proof theory, and it allows to draw meaningful conclusions from (potentially) conflicting and incomplete knowledge base. In non-monotonic systems, more accurate conclusions can be obtained when more pieces of information become available.

Many variants of S DL have been proposed for the logical modelling of different application areas, specifically agents [47,37,18], legal reasoning [41,17], and workflows from a business process compliance perspective [36,67,65].

We focus on this research on the deontic framework proposed by the authors in [35]: Defeasible Deontic Logic (DDL) allows us to determine which prescriptive behaviours are in force.

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). We use lower-case Roman letters to denote literals, and lower-case Greek letters to denote rules.

Accordingly, $\mathrm{PLit} = \mathrm{PROP} \cup \{\neg l \mid l \in \mathrm{PROP}\}$ is the set of *plain literals*. The set of *deontic literals* $\mathrm{ModLit} = \{\Diamond l, \neg\Diamond l \mid l \in \mathrm{PLit} \wedge \Diamond \in \{\mathrm{O}, \mathrm{P}\}\}$.[8] Finally, the set of *literals* is $\mathrm{Lit} = \mathrm{PLit} \cup \mathrm{ModLit}$. 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 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 $F \subseteq \mathrm{PLit}$ denotes simple pieces of information that are considered always to be true, like "Sylvester is a cat", formally $cat(Sylvester)$. In this paper we subscribe to distinction between the notions of obligations and permissions, and that of norms, where the obligations and permissions in force in a normative system are determined by the norms in the 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 correspond to a case. As we will see below, the rules are used to conclude what are the institutional facts, obligations and permissions that holds in case. Accordingly, we do not admit obligations and permissions to be 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);

[8] In this paper, we will use three modal operators C, O and P. We adopt the convention to use $\Box$ as a variable for any modality and $\Diamond$ when we refer to a deontic modality (O or P).

– *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[9]

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

as there is no exception to it[10].

On the other hand, defeasible rules are to conclude statements that can be defeated by contrary evidence the statement, whereas 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

$$\mathit{CScientist}, \mathit{PaperAccepted} \Rightarrow_{\mathrm{C}} \mathit{TravelConference}$$

$$\mathit{Pandemic} \rightsquigarrow_{\mathrm{C}} \neg \mathit{TravelConference}.$$

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

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

and the exception through the permissive rule

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

As we alluded to above [35] discusses how to integrate strong and weak permission in defeasible deontic logic. In this paper we focus on the notion of strong permission, namely, when permissions are explicitly stated by means of permissive rules, i.e., rules whose conclusion is to be asserted as a permission.

Following the ideas of [39], obligation rules gain more expressiveness with the *compensation operator* $\otimes$ for obligation rules, which is to model reparative chains of obligations. Intuitively, $a \otimes b$ means that a is the primary obligation, but if for some reason we fail to obtain, to comply with, a (by either not being able to prove a, or by proving $\sim a$) then b becomes the new obligation in force. This operator is used to build chains of preferences, called $\otimes$-expressions.

[9] Here, we introduce informally the symbols to represent different types of rules, which are formally defined below in Definition 2, where $\rightarrow$ denotes a strict rule, $\Rightarrow$ for a defeasible rule, and $\rightsquigarrow$ for a defeater.

[10] Like in [3], we consider only a propositional version of this logic, and we do not take into account function symbols. Every expression with variables represents the finite set of its variable-free instances.

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 [35]. ~~In addition, we stipulate that $\otimes$ obeys the following properties: (a) Associativity $a \otimes (b \otimes c) = (a \otimes b) \otimes c$, (b) Duplication and contraction on the right $\bigotimes_{i=1}^{m} c_i = (\bigotimes_{i=1}^{k-1} c_i) \otimes (\bigotimes_{i=k+1}^{m} c_i)$ for $j < k$ and $c_j = c_k$.~~

[**Gui 1**] Non vengono usate nel paper, si puo' mettere un referenza al/i paper(s) dove di mostrano le proprieta' di $\otimes$

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 illicit U-turn.

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

In this paper we focus exclusively on the defeasible part of the logic ignoring the monotonic component given by the strict rules; consequently, we limit the language to the cases where the rules are either defeasible or defeaters. From a practical point of view the restriction does not effectively limit the expressive power of the logic: a defeasible rule where there are no rules for the opposite conclusion, or where all rules for the opposite conclusion are weaker than the given defeasible rules, effectively behaves like a strict rule. Formally a rule is defined as below.

Definition 2 (Rule). *A* rule *is an expression of the form* $\alpha\colon A(\alpha) \hookrightarrow_{\square} C(\alpha)$, *where*

1. $\alpha \in \mathrm{Lab}$ *is the unique name of the rule;*
2. $A(\alpha) \subseteq \mathrm{Lit}$ *is the set of antecedents;*
3. *An arrow* $\hookrightarrow \in \{\Rightarrow, \rightsquigarrow\}$ *denoting, respectively, defeasible rules, and defeaters;*
4. $\square \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$*;*
5. *its consequent* $C(\alpha)$*, which is either*
   (a) *a single plain literal* $l \in \mathrm{PLit}$*, if either (i)* $\hookrightarrow \equiv \rightsquigarrow$ *or (ii)* $\square \in \{\mathrm{C}, \mathrm{P}\}$*, or*
   (b) *an* $\otimes$*-expression, if* $\square \equiv \mathrm{O}$*.*

If $\square = \mathrm{C}$ then the rule is used to derive non-deontic literals (constitutive statements), whilst if $\square$ is O or P then the rule is used to derive deontic conclusions (prescriptive statements). The conclusion $C(\alpha)$ is, as before, a single literal in case $\square = \{\mathrm{C}, \mathrm{P}\}$; in case $\square = \mathrm{O}$, then the conclusion is an $\otimes$-expression.

The meaning of an $\otimes$-expression '$C(\alpha) = c_1 \otimes c_2 \otimes \cdots \otimes c_m$' as consequent of a rule '$A(\alpha) \hookrightarrow_{\mathrm{O}} C(\alpha)$' is that: 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. 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. Note that we do not admit $\otimes$-chains on defeaters (Condition 5.(a).i), see [35] for a detailed explanation.

We use some abbreviations on sets of rules. The set of defeasible rules in R is $R_{\Rightarrow}$, the set of defeaters is R_{dft}. $R^{\square}[l]$ is the rule set appearing in R with head

l and modality $\Box$, while $R^{\mathrm{O}}[l, i]$ denotes the set of obligation rules where l is the i-th element in the $\otimes$-expression. Given that the consequent of a rule is either a single literal or an $\otimes$-expression (that, due to the associative property, can be understood as a sequence of elements, and then as an ordered set), in what follows we are going to abuse the notation and use $l \in C(\alpha)$. $R^{\Box}$ is the set of rules $\alpha\colon A(\alpha) \hookrightarrow_{\Box} C(\alpha)$ such that α *appears in* R. For a theory as determined by Definitions 1 and 2, α appears in R means that $\alpha \in R$. Finally, a literal l appears in a theory D, there is a rule $\alpha \in R$ such that $l \in A(\alpha) \cup C(\alpha)$.

Definition 3 (Tagged modal formula). *A* tagged modal formula *is an expression of the form $\pm\partial_{\Box} l$, with the following meanings*

- $+\partial_{\Box} l$*: l is* defeasibly provable *(or simply provable) with mode $\Box$,*
- $-\partial_{\Box} l$*: l is* defeasibly refuted *(or simply refuted) with mode $\Box$,*

that means l is defeasibly provable *(resp.* refuted*) in D as constitutive statement, obligation, or permission depending on $\Box$.*

[Gui 2] Magari possiamo tagliare la frase, o cambiarla completamente

As we will shortly see (Definitions 5 and 6) one of the key idea 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). Thus, when we have assert the tagged modal formula $+\partial_{\mathrm{O}} l$ in a derivation (see Definition 4 below), we are able to conclude that the obligation of l ($\mathrm{O}l$) follows from the rules and the facts, and that we used a prescriptive rule to derive l; 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 $+\partial_{\mathrm{C}} l$ the derivation simply asserts that l holds (and not that $\mathrm{C}l$ holds, even it the two have the same meaning). For the negative cases (i.e., $-\partial_{\Box} l$), the interpretation is that it is not possible to derive l with a given mode. Accordingly, we read $-\partial_{\mathrm{O}} l$ as it is not possible to derive l as an obligation. For $\Box \in \{\mathrm{O}, \mathrm{P}\}$ 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 $+\partial_{\mathrm{C}} \sim l$); in the logic failing to prove l does not equate to prove $\neg l$.

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

The definition of proof is also the standard in DDL.

Definition 4 (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 defined in the rest of this paper hold.*

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

Core notions in DL are that of *applicability/discardability* and of *team defeat*. (We shall discuss about team defeat after having introduced the proof tags.)

As knowledge in a defeasible theory is circumstantial, given a defeasible rule like '$\alpha\colon a, b \Rightarrow_{\Box} c$', there are four possible scenarios: the theory defeasibly proves both a and b, the theory proves neither, the theory proves one but not the other. Naturally, only in the first case, where both a and b are proved, we can

use α to *support/try to conclude* $\Box c$. Briefly, we say that a rule is *applicable* when every antecedent's literal has been proved at a previous derivation step. Symmetrically, a rule is *discarded* when one of such literals has been previously refuted. Formally:

Definition 5 (Applicability). *Assume a deontic defeasible theory* $D = (F, R, >)$. *We say that rule* $\alpha \in R^{\mathrm{C}} \cup R^{\mathrm{P}}$ *is* applicable *at* $P(n+1)$, *iff for all* $a \in A(\alpha)$

1. *if* $a \in \mathrm{PLit}$, *then* $+\partial_{\mathrm{C}} a \in P(1..n)$,
2. *if* $a = \Diamond q$, *then* $+\partial_{\Diamond} q \in P(1..n)$, *with* $\Diamond = \{\mathrm{O}, \mathrm{P}\}$,
3. *if* $a = \neg\Diamond q$, *then* $-\partial_{\Diamond} q \in P(1..n)$, *with* $\Diamond = \{\mathrm{O}, \mathrm{P}\}$.

We say that rule $\alpha \in R^{\mathrm{O}}$ *is* applicable at index i and $P(n+1)$ *iff Conditions 1–3 above hold and*

4. $\forall c_j \in C(\alpha)$, $j < i$, *then* $+\partial_{\mathrm{O}} c_j \in P(1..n)$ *and* $+\partial_{\mathrm{C}} {\sim} c_j \in P(1..n)$[11].

Definition 6 (Discardability). *Assume a deontic defeasible theory* D, *with* $D = (F, R, >)$. *We say that rule* $\alpha \in R^{\mathrm{C}} \cup R^{\mathrm{P}}$ *is* discarded *at* $P(n+1)$, *iff there exists* $a \in A(\alpha)$ *such that*

1. *if* $a \in \mathrm{PLit}$, *then* $-\partial_{\mathrm{C}} l \in P(1..n)$, *or*
2. *if* $a = \Diamond q$, *then* $-\partial_{\Diamond} q \in P(1..n)$, *with* $\Diamond = \{\mathrm{O}, \mathrm{P}\}$, *or*
3. *if* $a = \neg\Diamond q$, *then* $+\partial_{\Diamond} q \in P(1..n)$, *with* $\Diamond = \{\mathrm{O}, \mathrm{P}\}$

We say that rule $\alpha \in R^{\mathrm{O}}$ *is* discarded at index i and $P(n+1)$ *iff either at least one of the Conditions 1–3 above does not hold, or*

4. $\exists c_j \in C(\alpha)$, $j < i$ *such that* $-\partial_{\mathrm{O}} c_j \in P(1..n)$, *or* $-\partial_{\mathrm{C}} {\sim} c_j \in P(1..n)$.

Note that discardability is obtained by applying the principle of *strong negation* to the definition of applicability. The strong negation principle applies the function that simplifies a formula by moving all negations to an inner most position in the resulting formula, replaces the positive tags with the respective negative tags, and the other way around see [38]. Positive proof tags ensure that there are effective decidable procedures to build proofs; strong negation principle guarantees that the negative conditions provide a constructive and exhaustive method to verify that a derivation of the given conclusion is not possible.

We are finally ready to formalise the proof conditions, which are the standard in DDL [35]. We start with positive proof conditions for constitutive statements. In the following, we shall omit the explanations for negative proof conditions, when trivial, reminding the reader that are obtained through application of the strong negation principle to the positive counterparts.

[11] 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) that 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 note that the former option implies the latter one. For a deeper discussion on the issue, see [32].

Definition 7 (Constitutive Proof Conditions).

$+\partial_{\mathrm{C}} l$: *If* $P(n+1) = +\partial_{\mathrm{C}} l$ *then*
(1) $l \in F$, *or*
(2) (1) $\sim l \notin F$, *and*
(2) $\exists \beta \in R^{\mathrm{C}}_{\Rightarrow}[l]$ *s.t.* β *is applicable, and*
(3) $\forall \gamma \in R^{\mathrm{C}}[\sim l]$ *either*
(1) γ *is discarded, or*
(2) $\exists \zeta \in R^{\mathrm{C}}[l]$ *s.t.*
(1) ζ *is applicable and*
(2) $\zeta > \gamma$.

$-\partial_{\mathrm{C}} l$: *If* $P(n+1) = -\partial_{\mathrm{C}} l$ *then*
(1) $l \notin F$ *and either*
(2) (1) $\sim l \in F$, *or*
(2) $\forall \beta \in R^{\mathrm{C}}_{\Rightarrow}[l]$, *either* β *is discarded, or*
(3) $\exists \gamma \in R^{\mathrm{C}}[\sim l]$ *such that*
(1) γ *is applicable, and*
(2) $\forall \zeta \in R^{\mathrm{C}}[l]$, *either* ζ *is discarded, or* $\zeta \not> \gamma$.

A literal is defeasibly proved if: it is a fact, or there exists an applicable, defeasible rule supporting it (such a rule cannot be a defeater), and all opposite rules are either discarded, or defeated. To prove a conclusion, not all the work has to be done by a stand-alone (applicable) rule (the rule witnessing condition (2.2): all the applicable rules for the same conclusion (may) contribute to defeat applicable rules for the opposite conclusion. Note that both γ as well as ζ may be defeaters.

Example 1. Let $D = (F = \{a, b, c, d, e\}, R, > = \{(\alpha, \varphi), (\beta, \psi)\})$ be a theory such that

$$R = \{\alpha\colon a \Rightarrow_{\mathrm{C}} l \qquad \beta\colon b \Rightarrow_{\mathrm{C}} l \qquad \gamma\colon c \Rightarrow_{\mathrm{C}} l$$
$$\varphi\colon d \Rightarrow_{\mathrm{C}} \neg l \qquad \psi\colon e \Rightarrow_{\mathrm{C}} \neg l \qquad \chi\colon g \Rightarrow_{\mathrm{C}} \neg l\}.$$

Here, $D \vdash +\partial_{\mathrm{C}} f_i$, for each $f_i \in F$ and, by Condition (1) of $+\partial$. Therefore, all rules but χ (which is discarded) are applicable: χ is indeed discarded since no rule has g as consequent nor is a fact. The team defeat supporting l is made by α, β and γ, whereas the team defeat supporting $\neg l$ is made by φ and ψ. Given that α defeats φ and β defeats ψ, then we conclude that $D \vdash +\partial_{\mathrm{C}} l$. Note that, despise being applicable, γ does not effectively contribute in proving $+\partial_{\mathrm{C}} l$, i.e. D without γ would still prove $+\partial_{\mathrm{C}} l$.

Suppose to change D such that both α and β are defeaters. Even if γ defeats neither φ nor ψ, γ is now needed to prove $+\partial l$ as Condition (2.2) requires that at least one applicable rule must be a defeasible rule. Below we present the proof conditions for obligations.

Definition 8 (Obligation Proof Conditions).

$+\partial_{\mathrm{O}} l$: *If* $P(n+1) = +\partial_{\mathrm{O}} l$ *then*
$\exists \beta \in R^{\mathrm{O}}_{\Rightarrow}[l, i]$ *s.t.*
(1) β *is applicable at index* i *and*
(2) $\forall \gamma \in R^{\mathrm{O}}[\sim l, j] \cup R^{\mathrm{P}}[\sim l]$ *either*
(1) γ *is discarded (at index* j*), or*
(2) $\exists \zeta \in R^{\mathrm{O}}[l, k]$ *s.t.*
(1) ζ *is applicable at index* k *and*
(2) $\zeta > \gamma$.

$-\partial_{\mathrm{O}} l$: *If* $P(n+1) = -\partial_{\mathrm{O}} l$ *then*
$\forall \beta \in R^{\mathrm{O}}_{\Rightarrow}[l, i]$ *either*
(1) β *is discarded at index* i*, or*
(2) $\exists \gamma \in R^{\mathrm{O}}[\sim l, j] \cup R^{\mathrm{P}}[\sim l]$ *s.t.*
(1) γ *is applicable (at index* j*), and*
(2) $\forall \zeta \in R^{\mathrm{O}}[l, k]$ *either*
(1) ζ *is discarded at index* k*, or*
(2) $\zeta \not> \gamma$.

Note that: (i) in Condition (2) γ can be a permission rule as explicit, opposite permissions represent exceptions to obligations, whereas ζ (Condition 2.2) must be an obligation rule as a permission rule cannot reinstate an obligation, and that (ii) l may appear at different positions (indices i, j, and k) within the three $\otimes$-chains. The example below supports the intuition behind the restriction to obligation rules in Conditions (2.2).

Example 2. Suppose that medical guideline of a hospital forbid the use of opiates to sedate patients with an addiction history. However, at the same time, the guidelines mandate the same kind of drugs for terminal patients with cancer. However, physicians are permitted to refute to treat patients with opioids based on moral ground objections.

$$\alpha\colon \mathit{AddictionHistory} \Rightarrow_{\mathrm{O}} \neg \mathit{Opioids}$$
$$\beta\colon \mathit{TerminalCancer} \Rightarrow_{\mathrm{O}} \mathit{Opioids}$$
$$\gamma\colon \mathit{MoralGround} \Rightarrow_{\mathrm{P}} \neg \mathit{Opioids}$$

where $\gamma > \beta$ and $\beta > \alpha$. Is it forbidden to use opiates in case of a terminally ill patient with addition history where moral ground objections apply? Here, rule γ establishes an exemption from the obligation to prescribe opiates, however the opposite course of action is admissible, and the use of opiates for terminally ill cancer patients appears to be not forbidden (with or without moral ground objections).

Below we introduce the proof conditions for permissions.

Definition 9 (Permission Proof Conditions).

$+\partial_{\mathrm{P}} l$: *If* $P(n+1) = +\partial_{\mathrm{P}} l$ *then*
(1) $+\partial_{\mathrm{O}} l \in P(1..n)$*, or*

(2) $\exists \beta \in R^{\mathrm{P}}_{\Rightarrow}[l]$ *s.t.*
(1) β *is applicable and*
(2) $\forall \gamma \in R^{\mathrm{O}}[\sim l, j]$ *either*
(1) γ *is discarded at index* j*, or*
(2) $\exists \zeta \in R^{\mathrm{P}}[l] \cup R^{\mathrm{O}}[l, k]$ *s.t.*
(1) ζ *is applicable (at index* k*) and*
(2) $\zeta > \beta$.

$-\partial_{\mathrm{P}} l$*: If* $P(n+1) = -\partial_{\mathrm{P}} l$ *then*
(1) $-\partial_{\mathrm{O}} l \in P(1..n)$*, and*
(2) $\forall \beta \in R^{\mathrm{P}}_{\Rightarrow}[l]$ *either*
(1) β *is discarded or*
(2) $\exists \gamma \in R^{\mathrm{O}}[\sim l, j]$ *s.t.*
(1) γ *is applicable at index* j *and*
(2) $\forall \zeta \in R^{\mathrm{P}}[l] \cup R^{\mathrm{O}}[l, k]$ *either*
(1) ζ *is discarded (at index* k*), or*
(2) $\zeta \not> \beta$.

Condition (1) allows us to derive a permission from the corresponding obligation, thus it corresponds to the $\mathrm{O}a \rightarrow \mathrm{P}a$ axiom of Deontic Logic.~~weak permissions as the absence of the opposite obligation~~. Condition (2.2) considers as possible counter-arguments *only* obligation rules as situations where both $\mathrm{P}l$ and $\mathrm{P}{\sim}l$ hold are allowed. We refer the readers interested in a deeper discussion on how to model permissions and obligations in DDL to [35].

[Gui 3] Vogliamo dire qualcosa su weak permission? Thus that weak permissions is $-\partial_{\mathrm{O}} \neg$?

Hereafter, whenever the applicability conditions of a given rule are not relevant for the example, we will set the corresponding set of antecedents to empty as such rules are vacuously applicable.

Example 3. Assume the theory of Example 1, where we extend the rule set and the superiority relation as follows

$$R \cup \{\zeta\colon \emptyset \Rightarrow_{\mathrm{O}} \neg l \otimes p \quad \eta\colon \emptyset \Rightarrow_{\mathrm{P}} l \quad \nu\colon \neg \mathrm{O} l \Rightarrow_{\mathrm{C}} q\}$$
$$> \cup \{(\zeta, \eta)\}.$$

Since $\zeta > \eta$, we conclude $D \vdash +\partial_{\mathrm{O}} \neg l$, $D \vdash -\partial_{\mathrm{P}} l$ due to Condition (3) of $-\partial_{\mathrm{P}}$, and $D \vdash -\partial_{\mathrm{O}} l$ due Condition (1) of $-\partial_{\mathrm{O}}$ as there are no obligation rules supporting l. Condition (3) of Definition 5 is satisfied, which makes ν applicable and hence $D \vdash +\partial_{\mathrm{C}} q$.

Given that, from Example 1, $D \vdash +\partial_{\mathrm{C}} l$, Condition (4) of Definition 5 is true which makes ζ applicable at index 2 for p: since the are no deontic rules supporting either $\mathrm{O}{\sim}p$ or $\mathrm{P}{\sim}p$, we conclude that $D \vdash +\partial_{\mathrm{O}} p$.

The set of positive and negative conclusions of a theory is called *extension*. The extension of a theory is computed based on the literals that appears in it; more precisely, the literals in the Herbrand Base of the theory $HB(D) = \{l, {\sim}l \in \mathrm{PLit} \mid l \text{ appears in } D\}$.

Definition 10 (Extension). *Given a defeasible, deontic theory* D*, we define the* extension *of* D *as*

$$E(D) = (+\partial_{\mathrm{C}}, -\partial_{\mathrm{C}}, +\partial_{\mathrm{O}}, -\partial_{\mathrm{O}}, +\partial_{\mathrm{P}}, -\partial_{\mathrm{P}}),$$

where $\pm\partial_{\square} = \{l \in HB(D) | \, D \vdash \pm\partial_{\square} l\}$, *with* $\square \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$.

In Example 3, $E(D)$ consists of the following sets, for each $f_i \in F$,

$$+\partial_{\mathrm{C}} = \{f_i, l, q\}, \qquad -\partial_{\mathrm{C}} = \{\neg f_i, \neg l, \neg q\},$$
$$+\partial_{\mathrm{O}} = \{\neg l, p\}, \qquad -\partial_{\mathrm{O}} = \{f_i, \neg f_i, l, q, \neg q\}$$
$$+\partial_{\mathrm{P}} = \{\neg l, p\}, \qquad -\partial_{\mathrm{P}} = \{f_i, \neg f_i, l, \sim p, q, \neg q\}).$$

5 Defeasible Deontic Logic with Meta-Rules

Many variants of Standard DL have been proposed for the logical modelling of different application areas, specifically agents [47,37,18], legal reasoning [41,17], and workflows from a business process compliance perspective [36,67,65].

Some of these application fields require the modelling of contexts and the use of rules in the scope of other rules. Accordingly, extensions of the logic have been developed to capture such features by adopting meta-rules. However, the work on meta-rules in Defeasible Logic focused on defining the extensions of the logic, specifically the proof theoretic features, neglecting to investigate the computational aspects.

In a recent work, the authors of [66] have been the first, to the best of our knowledge, to provide a first complete logical framework that encodes meta-rules within a non-monotonic logic.

This paper generalises [66]'s model by

- Adding deontic operators for obligation (O) and permission (P);
- Adding the compensation operator $\otimes$;
- Applying deontic and compensation operators both to literals and to rules;
- Deeply investing situations where rules would provide conflicting information wrt other rules.

The richness of this extension makes the logic very expressive and leads to examining a potentially large variety of conflicts between meta-rules. Consider the following meta-rules:

$$\zeta\colon A(\zeta) \Rightarrow_{\square} (\alpha)$$
$$\nu\colon A(\nu) \Rightarrow_{\square} (\beta).$$

On the one hand, we can simply extend to rules the intuition covering literals: as literal s conflicts with another literal l whenever s is the complement of l ($\sim l$, which is $\neg b$ if $b = l$), rule ν conflicts with rule ζ whenever $\beta = \sim\alpha$, i.e. $\beta = \neg\alpha$.

What does it mean applying $\neg$ to a rule? In our context – normative reasoning – if a rule α corresponds to a norm, then $\neg\alpha$ means that such a norm does not exist in the normative system, or that it is removed.

Example 4. Consider this case from Italian Law. The Parliament issues the following Legislative Act (n. 124, 23 July 2008):

> [**Target of the modification**] With the exception of the cases mentioned under the Articles 90 and 96 of the Constitution, criminal proceedings against the President of the Republic, the President of the Senate,

> the President of the House of Representatives, and the Prime Minister, are suspended for the entire duration of tenure. [...]

Suppose the Constitutional Court declares that the legislative act above is illegitimate and states its annulment. This means that such a norm no longer exists in the system. Hence, if issuing the norm corresponds to

$$\zeta\colon \mathit{Parliament}, \mathit{Promulgation} \Rightarrow_C (\mathit{L.\ 124}\colon \mathit{Crime}, \mathit{Tenure} \Rightarrow_{\mathrm{O}} \mathit{Suspended})$$

then the annulment of *L. 124* can be reconstructed by having, in the meta-theory, a meta-rule such as

$$\nu\colon \mathit{Constitutional\_Court} \Rightarrow_C \neg(\mathit{L.\ 124}\colon \mathit{Crime}, \mathit{Tenure} \Rightarrow_{\mathrm{O}} \mathit{Suspended}).$$

Clearly, ν conflicts with ζ.

If conflicts between meta-rules occur only in cases like this, our variant of the meta-logic extends the standard intuitions of rule conflict in DL to meta-rules. On the basis of the discussion in Section 3, we call this variant *simple*.

However, we can inspect the deontic meaning of rules occurring in the head of meta-rules, an inspection that allows us to introduce a second variant of the meta-logic. Again on the basis of the discussion in Section 3, this variant, which we call *prudential*, takes into account the following scenarios of when meta-rules are in conflict.

Case 1. Consider the following meta-rules:

$$\zeta\colon A(\zeta) \Rightarrow_\Box (\alpha\colon a \Rightarrow_\Box b)$$
$$\nu\colon A(\nu) \Rightarrow_\Box (\beta\colon a \Rightarrow_\Box \neg b)$$

The two meta-rules can be seen as incompatible (especially when $\Rightarrow_\Box$ is $\Rightarrow_{\mathrm{O}}$), since rules α and β have exactly the same antecedent, they are labelled by the same $\Box$-modality, but support complementary conclusions. A variant is

$$\zeta\colon A(\zeta) \Rightarrow_\Box (\alpha\colon a \Rightarrow_{\mathrm{O}} b)$$
$$\nu\colon A(\nu) \Rightarrow_\Box (\beta\colon a \Rightarrow_{\mathrm{P}} \neg b)$$

where the modalities of α and β are different (specifically, one is obligation and the other permission), but we have a conflict as well.

Case 2. Consider the following meta-rules:

$$\zeta\colon A(\zeta) \Rightarrow_\Box (\alpha\colon a \Rightarrow_{\mathrm{O}} b \otimes c)$$
$$\nu\colon A(\nu) \Rightarrow_\Box (\beta\colon a \Rightarrow_{\mathrm{O}} \neg b \otimes d).$$

This is somehow similar to the previous case, but (i) it focuses on the obligation rules, and (ii) the head of rules are $\otimes$-expressions. Again, α and β have exactly the same antecedent, but, although the heads state consistent compensations, their primary obligations are incompatible. Therefore, ζ and ν are in conflict because support the conclusions Ob and O$\neg b$.

Case 3. Consider the following meta-rules:

$$\zeta\colon A(\zeta) \Rightarrow_{\Box} (\alpha\colon a \Rightarrow_{\mathrm{O}} b \otimes d)$$
$$\nu\colon A(\nu) \Rightarrow_{\Box} (\beta\colon a \Rightarrow_{\mathrm{O}} c \otimes {\sim}d).$$

Following the analysis for Case 2 above but for opposite reasons: the two meta-rules are not in conflict. Rules α and β have exactly the same antecedent, but, although the heads state inconsistent compensations, their primary obligations are compatible. Ergo, ζ and ν are *not* in conflict because support the conclusions Ob and Oc.

Case 4. Consider the following meta-rules:

$$\zeta\colon A(\zeta) \Rightarrow_{\Box} (\alpha\colon a \Rightarrow_{\mathrm{O}} b)$$
$$\nu\colon A(\nu) \Rightarrow_{\Box} (\beta\colon a \Rightarrow_{\mathrm{O}} \neg b \otimes d).$$

This is similar to Case 2, hence ζ and ν are in conflict.

Case 5. Consider the following meta-rules:

$$\zeta\colon A(\zeta) \Rightarrow_{\Box} (\alpha\colon a \Rightarrow_{\mathrm{O}} b)$$
$$\nu\colon A(\nu) \Rightarrow_{\Box} (\beta\colon a \Rightarrow_{\mathrm{O}} b \otimes d).$$

Strictly speaking, these meta-rules are not in conflict as they support the derivation of rules both supporting Ob. On a different level – the compliance angle – we can argue that a theory containing both meta-rules is somehow odd, since β specifies that the violation of b is compensated by d; hence, a situation s satisfying a, $\neg b$ and c complies with β. On the contrary, α does not admit a compensation for the violation of b, and the situation s contravenes α. Therefore, a system where both α and β are in force admits a situation that is, at the same time, compliant and non compliant.

Case 6. Consider the following meta-rules:

$$\zeta\colon A(\zeta) \Rightarrow_{\Box} (\alpha\colon a \Rightarrow_{\mathrm{O}} b \otimes d)$$
$$\nu\colon A(\nu) \Rightarrow_{\Box} (\beta\colon a \Rightarrow_{\mathrm{O}} b \otimes \neg d)$$

Following the analysis for other cases, the two meta-rules are in conflict. Rules α and β have exactly the same antecedent and prove the same primary obligation, but they also state inconsistent compensations. In fact, if Ob is violated, α and β support opposite conclusions: Od and O$\neg d$.

Case 7. Consider the following meta-rules:

$$\zeta\colon A(\zeta) \Rightarrow_{\Box} (\alpha\colon a \Rightarrow_{\mathrm{O}} b \otimes c)$$
$$\nu\colon A(\nu) \Rightarrow_{\Box} (\beta\colon a \Rightarrow_{\mathrm{O}} b \otimes \neg d).$$

The two meta-rules are not in conflict, even though they can be seen as deontically odd: the 'apparent oddity' lies in the fact that the same violation of Ob leads to different compensations.

To further illustrate the intuitions behind the prudential meta-logic's perspective, consider this second legal example.

Example 5. Consider Art. 3 of the Italian Constitution:

> Article 3
> All citizens have equal social status and are equal before the law, without regard to their sex, race, language, religion, political opinions, and personal or social conditions.

Suppose the Constitution is amended by stating the following:

> In derogation to the provisions set out in Article 3, paragraph 1, of the Constitution, when Italy will no longer be a member state of the EU, the EU citizens may have different social status.

If Art. 3 can be represented as follows

$$Art.\ 3\colon Citizen \Rightarrow_{\mathrm{O}} Equal\_status$$

its amendment corresponds to applying the following meta-rule:

$$derog_{Art.\ 3}\colon \neg EU \Rightarrow_{C} (\mu\colon EU\_Citizen \Rightarrow_{\mathrm{P}} \sim Equal\_status).$$

The example corresponds to the second variant of Case 1 above. Similar examples can be elaborated to illustrate the other scenarios previously discussed.

5.1 Basics

In the new language we accommodate both rules and meta-rules, where a meta-rule is a rule whose elements are other rules. We are now providing the formal machinery to properly capture such notions. The main idea is to re-use as much as possible the definitions and constructions given in Section 4, and to revise/extend them to include the new notions in this way the variants of the Defeasible Deontic Logic with meta-rules we are going to present in this Section are a conservative extension of the logic of Section 4. Accordingly, for instance, there is not need to give a new definition of what a derivation is, or to adjust the conditions when literals and deontic literals are provable.

For the terminology used, we reserve the term *standard rules* for expressions satisfying Definition 2 (rules containing only literals), and we call *meta-rules* expressions containing other (standard) rules. Notice that we do not allow for the nesting of meta-rules.[Gui]

N.B. For the current investigation we do not For the terminology used, we reserve the term *standard rules* for expressions satisfying Definition 2 (rules containing only literals), and we call *meta-rules* expressions containing standard rules, *but not other meta-rules. Notice that we do not allow for the nesting of meta-rules.*.

~~For the terminology used, we reserve the term *standard rules* for expressions satisfying Definition 2 (rules containing only literals), and we call *meta-rules* expressions containing other (standard) rules.~~[Ce]

In most cases, we do not need to distinguish between standard and meta-rules: Definition 12 captures both into a single definition of what a rule is (when we speak of a generic rule, it can be thus be either).

Definition 11 (Rule Expressions). *Given a standard rule α, α and $\neg\alpha$ are* rule expressions*; we use* REx *to denote the set of rule expressions. If β is a rule expression, then $\Diamond\beta$ and $\neg\Diamond\beta$ are* deontic rule expressions*.*

We also extend the definition of $\otimes$-expression allowing every element to be either a plain literal, or a rule expression (see the example below).

Example 6. The following is an $\otimes$-expression in the DDLmR language

$$a \otimes (\beta\colon b \Rightarrow_{\mathrm{O}} c) \otimes (\gamma\colon d, \mathrm{O}a \Rightarrow_{\mathrm{O}} e \otimes \neg f).$$

Note that we can mix rules and literals in reparation chains. This allows us to represent situations where, for instance, an entity needs to enforce a particular policy, and being subject to a sanction if it does not.

Consequently, we redefine the notion of rule as follows.

Definition 12 (Rule). *A* rule *is an expression $\alpha : A(\alpha) \hookrightarrow_{\square} C(\alpha)$, where*

1. *$\alpha \in$ Lab is the unique name of the rule.*
2. *$A(\alpha)$ is a (possibly empty) set of elements $a_1, \ldots, a_n$, where each a_i is either a literal, a rule expression, or a deontic rule expression;*
3. *$\hookrightarrow \in \{\Rightarrow, \rightsquigarrow\}$ with the same meaning as before;*
4. *$\square = \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$;*
5. *its consequent $C(\alpha)$, which is either*
   (a) *a single plain literal $l \in$ PLit, or a rule expression β, if either (i) $\hookrightarrow \equiv \rightsquigarrow$ or (ii) $\square \in \{\mathrm{C}, \mathrm{P}\}$, or*
   (b) *an $\otimes$-expression, if $\square \equiv \mathrm{O}$.*

We extend the convention used to represent the complement of a literal to the cases of (deontic) rule expressions: (i) if $\beta = \alpha$ then $\sim\beta = \neg\alpha$, and (ii) if $\beta = \neg\alpha$ then $\sim\beta = \alpha$.

Definition 12 allows us to reuse the notation specified to identify particular sets of rules given in Section 4. Here, we extend the terminology and we say that a rule α *appears in a rule* β, if $\alpha \in A(\beta) \cup C(\beta)$, and that α *appears in a set of rules* S, if either $\alpha \in S$, or $\exists \beta \in S$ such that α appears in β.

As standard rules may now be conclusions of meta-rules, we have to update the definition of tagged modal formula given in Definition 3 as follows.

Definition 13 (Tagged modal formula). *An expression is a* tagged modal formula *if satisfies Definition 3, or has one of the following forms:*

- *$+\partial^m_{\square}\alpha$: meaning that rule α is defeasibly provable with mode $\square$;*
- *$-\partial^m_{\square}\alpha$: meaning that rule α is defeasibly refuted with mode $\square$.*

The intuition here is the same as for literals. In a defeasible meta-theory, the conclusion of a derivation can be either a (deontic) literal, or a (deontic) rule. Accordingly, when a theory proves $+\partial^m_{\mathrm{O}}\alpha$, it is obligatory to have the rule α; when we prove a rule with mode C, the meaning is that the rule is in the system and can produce its conclusion; $-\partial^m_{\mathrm{C}}\alpha$, on the other hand, indicates that rule α is not present in the underlying normative system.

What about the case when we have the negation of a rule, $\neg\alpha$? For $+\partial^m_{\mathrm{C}}\neg\alpha$ two interpretations are possible: (i) an affirmative statement that α is not in the system, and (ii) that α has been removed from the rules of the system. The latter reading is what was used in [41] to model abrogation and annulment. For O, a prohibition is defined as O$\neg$. Accordingly, $+\partial^m_{\mathrm{O}}\neg\alpha$ specifies that, in the normative system, it is forbidden to have rule α. For instance, Article 27 of the Italian Constitution makes capital punishment not admissible in the Italian Legal System.

The definition of Proof does not change wrt Definition 4. However, we need to reformulate definitions of a rule being applicable/discarded to accommodate the case when a standard rule (or more properly a rule expression) is in the antecedent of a meta-rule. A meta-rule is applicable when (i) rule itself is provable and (ii) the rule expressions appearing in it are provable (or discarded) with the appropriate mode.

Definition 14 (Applicability). *Assume a deontic defeasible meta-theory D with $D = (F, R, >)$. We say that rule $\alpha \in R^{\mathrm{C}} \cup R^{\mathrm{P}}$ is* applicable *at $P(n+1)$, iff $+\partial^m_{\mathrm{C}}\alpha \in P(1..n)$ and for all $a \in A(\alpha)$ Conditions 1–3 of Definition 5 hold and*

1. *if $a = \beta \in$ REx, then $+\partial_{\mathrm{C}}\beta \in P(1..n)$,*
2. *if $a = \Diamond\beta$, $\beta \in$ REx, then $+\partial_{\Diamond}\beta \in P(1..n)$,*
3. *if $a = \neg\Diamond\beta$, $\beta \in$ REx, then $-\partial_{\Diamond}\beta \in P(1..n)$.*

We say that rule $\alpha \in R^{\mathrm{O}}$ is applicable at index i and *$P(n+1)$ iff $+\partial^m_{\mathrm{C}}\alpha \in P(1..n)$ and Conditions 1–3 of Definition 5 and Conditions 1-3 above hold, and for all $c_j \in C(\alpha)$, $j < i$*

4. a. *if $c_j = l$, then $+\partial_{\mathrm{O}}l \in P(1..n)$ and $+\partial_{\mathrm{C}}{\sim}l \in P(1..n)$, $l \in$ PLit;*
   b. *if $c_j = \beta$, then $+\partial_{\mathrm{O}}\beta \in P(1..n)$ and $-\partial_{\mathrm{C}}\beta \in P(1..n)$.*

Note that, whilst 4.a is the same of Condition 4 of Definition 4, when c_j is a rule then we require in Condition 4.b only $-\partial_{\mathrm{C}}\beta$ and not $+\partial_{\mathrm{C}}{\sim}\beta$. Both conditions indicate that rule β is not a rule effective. As discussed in Footnote 11, the former means that there is no evidence that the rule is in the system, while the latter guarantees that the rule is removed from or prevented to be in system. In a positivist view, a normative system explicitly gives the norms that hold in it. Consequently, the set of effective norms (the norms that can produce an effect) are constrained by the set of norms that appears in the normative system. We have argued that $-\partial^m_{\mathrm{C}}\alpha$ states that rule/norm α is not in the system (or that it is not effective). Therefore, this is enough to contravene the obligation of the norm being effective in the normative system.

Definition 15 (Discardability). *Assume a deontic defeasible theory D with $D = (F, R, >)$. We say that rule $\alpha \in R^{\mathrm{C}} \cup R^{\mathrm{P}}$ is* discarded *at $P(n+1)$, iff $-\partial^m_{\mathrm{C}}\alpha \in P(1..n)$ or at at least one of the Conditions 1–3 of Definition 6 does not hold, or there exists $a \in A(\alpha)$ such that*

1. *if $a = \beta \in$ REx, then $-\partial_{\mathrm{C}}\beta \in P(1..n)$,*
2. *if $a = \Diamond\beta$, $\beta \in$ REx, then $-\partial_{\Diamond}\beta \in P(1..n)$,*
3. *if $a = \neg\Diamond\beta$, $\beta \in$ REx, then $+\partial_{\Diamond}\beta \in P(1..n)$.*

We say that rule $\alpha \in R^{\mathrm{O}}$ *is* discarded at index i and $P(n+1)$ *iff* $-\partial^{m}_{\mathrm{C}}\alpha \in P(1..n)$ *or at least one of the Conditions 1–3 of Definition 6 or of the Conditions 1–3 above does not hold, or there exists* $c_j \in C(\alpha)$, $j < i$ *such that*

4. *a. if* $c_j = l$, *then* $-\partial_{\mathrm{O}} l \in P(1..n)$ *or* $-\partial_{\mathrm{C}} {\sim} l \in P(1..n)$, $l \in \mathrm{PLit}$*;*
   b. if $c_j = \beta$, *then* $-\partial_{\mathrm{O}}\beta \in P(1..n)$ *or* $+\partial_{\mathrm{C}}\beta \in P(1..n)$.

Before moving to the presentation of the two variants of the logic, we introduce a key feature of our logical apparatus: the notion of *conflict of rules*. The definition below formalises the intuitions discussed at the beginning of this section.

[**Gui 4**] aggiunto nuova definizione di conflitto semplice, rinominato la vecchia definizione rational conflict

Definition 16 (Simple Conflict). *Given two rules or rule expressions* α *and* β*, we say that* α *and* β simply conflict *iff*

1. $\alpha\colon A(\alpha) \hookrightarrow_{\Box} C(\alpha)$
   β *is* ${\sim}(\gamma\colon A(\alpha) \hookrightarrow_{\Box} C(\alpha))$
2. *If* $\nu \in C(\alpha)$ *at index* i, $\zeta \in C(\beta)$ *at index* j, *and* ν *and* ζ *simply conflict.*

Definition 17 (Prudential Conflict). *Given two rules or rule expressions* α *and* β*, we say that* α *and* β prudentially conflict *iff*

1. $\alpha\colon A(\alpha) \hookrightarrow_{\Box} C(\alpha)$
   β *is* $= {\sim}(\gamma\colon A(\alpha) \hookrightarrow_{\Box} C(\alpha))$
2. $\alpha\colon A(\alpha) \hookrightarrow_{\Box} C(\alpha)$,
   $\beta\colon A(\alpha) \hookrightarrow_{\Box} {\sim}C(\alpha)$*;*
3. $\alpha\colon A(\alpha) \hookrightarrow_{\mathrm{O}} C(\alpha)$,
   $\beta\colon A(\alpha) \hookrightarrow_{\mathrm{P}} {\sim}C(\alpha)$*;*
4. $\alpha\colon A(\alpha) \Rightarrow_{\mathrm{O}} c_1 \otimes \cdots \otimes c_m$,
   $\beta\colon A(\alpha) \Rightarrow_{\mathrm{O}} d_1 \otimes \cdots \otimes d_n$, *and*
   (a) $\exists (i \leq m, n) i \leq m \wedge i \leq n. \forall k < i.\, c_k = d_k \wedge c_i = {\sim}d_i$*; or*
   (b) $m < n \wedge \forall i \leq m\ c_i = d_i$.
5. *If* $\nu \in C(\alpha)$ *at index* i, $\zeta \in C(\beta)$ *at index* j, *and* ν *and* ζ *prudentially conflict.*

Given two rules or rule expressions α and β that (simply/prudentially) conflict, we will say that α (simply/prudentially) conflicts with β (and the other way around).[Gui] Notice that the definitions above apply to both standard rules and meta-rules; only the recursive conditions (condition 2 in Defintion 16 and condition 5 of Definition 17) are limited to meta-rules.

[**Gui 5**] Fare esempi di conflitti

Example 7.

In the rest of the section we shall introduce two variants of the logic, where in addition to prove literals we prove rules. The two variants share the proof conditions for literals given in Section 4 (Definitions 7–9), whilst they differ from each other in the proof conditions to derive (standard) rules.

5.2 Simple Conflict Defeasible Deontic Logic

We are now ready to present the proof conditions for the Simple Conflict variant of Defeasible Deontic Meta-Logic. In this variant, we focus on the case when there are applicable meta-rules for a standard rule and its negation. In other terms, when there are two meta-rules: one to introduce a rule in the system and one to remove it (or to prevent it to be inserted). To handle this case we restrict the logic to notion of simple conflicts.

Definition 18 (Constitutive Meta-Proof Conditions - Simple Variant).

$+\partial^m_{\mathrm{C}}\alpha$: *If* $P(n+1) = +\partial^m_{\mathrm{C}}\alpha$ *then*
(1) $\alpha \in R$ *or*
(2) (1) $\forall \omega \in R$, ω *does not simply conflict with* α, *and*
(2) $\exists \beta \in R^{\mathrm{C}}_{\Rightarrow}[(\alpha : A(\alpha) \hookrightarrow_{\Box} c)]$ *s.t.*
(1) β *is applicable, and*
(2) $\forall \gamma \in R^{\mathrm{C}}[\sim(\varphi : A(\alpha) \hookrightarrow_{\Box} c)]$ *either*
(1) γ *is discarded, or*
(2) $\exists \zeta \in R^{\mathrm{C}}[(\chi : A(\alpha) \hookrightarrow_{\Box} c)]$ *s.t.*
(1) $\chi \in \{\alpha, \varphi\}$,
(2) ζ *is applicable, and*
(3) $\zeta > \gamma$.

$-\partial^m_{\mathrm{C}}\alpha$: *If* $P(n+1) = -\partial^m_{\mathrm{C}}\alpha$ *then*
(1) $\alpha \notin R$ *and*
(2) (1) $\exists \rho \in R$, ρ *simply conflicts with* α, *or*
(2) $\forall \beta \in R^{\mathrm{C}}_{\Rightarrow}[(\alpha : A(\alpha) \hookrightarrow_{\Box} c)]$ *either*
(1) β *is discarded, or*
(2) $\exists \gamma \in R^{\mathrm{C}}[\sim(\varphi : A(\alpha) \hookrightarrow_{\Box} c)]$, *s.t.*
(1) γ *is applicable*
(2) $\forall \zeta \in R^{\mathrm{C}}[(\chi : A(\alpha) \hookrightarrow_{\Box} c)]$
(1) $\chi \notin \{\alpha, \varphi\}$, *or*
(2) ζ *is discarded, or*
(3) $\zeta \not> \gamma$.

For the explanation, we focus on the positive case. We first remark that the general structure of the proof conditions for rules is the same as that for literals. Either the conclusion is given in the theory, or there is an applicable rule for it and all possible attacks are either discarded or defeated. Ergo, the first option to prove a rule is to see if it is one to the rules given in R. Notice that this step applies to both standard rules and meta-rules (Condition 1). If rule α is not in R, Condition (2) checks that rules simply conflicting with α are not in the set of the initial, given rules R; if this step succeeds, we then look for an applicable meta-rule whose conclusion is α. It is worth noticing that, with the extended definition of applicability, β has to be provable, and this condition applies to the proof conditions for literals. Next, we have to examine all meta-rules for the "opposite" of α: here we consider all rule expressions starting with

a negation, and having the same antecedent as well as the same consequent of α; nevertheless, they can have a different label (we do not impose that label φ must be α). The reason being that it would be irrational to introduce a rule (with a given name) and at the same time to remove it (potentially with a different name). For all attacking meta-rules, we have to rebut them. As usual, there are two ways: either the attacking rule is discarded, or is defeated. However, contrary to the attacking phase, the defeating meta-rules must be for either α, or φ; in other terms, we cannot reinstate a specific rule using other generic rules.

Example 8. Consider the theory D where $a \in F$, the following rules are in R

$$\beta\colon \ldots \Rightarrow_{\mathrm{C}} (\alpha\colon a \Rightarrow_{\mathrm{C}} b) \quad \eta\colon \ldots \Rightarrow_{\mathrm{C}} c \quad \lambda\colon \ldots \rightsquigarrow_{\mathrm{C}} (\alpha\colon a \Rightarrow_{\mathrm{C}} b)$$
$$\gamma\colon c \Rightarrow_{\mathrm{C}} \neg(\epsilon\colon a \Rightarrow_{\mathrm{C}} b) \quad \theta\colon \ldots \Rightarrow_{\mathrm{C}} \neg(\eta\colon \ldots \Rightarrow_{\mathrm{C}} c) \quad \mu\colon d \Rightarrow_{\mathrm{C}} \neg(\alpha\colon a \Rightarrow_{\mathrm{C}} b)$$

and $(\lambda, \gamma) \in >$.

It is immediate to see that the meta-rules $\beta, \gamma, \theta, \lambda, \mu$ are provable being in R. Also we have $+\partial^m_{\mathrm{C}}\eta$, while there is an applicable meta-rule for $\neg\eta$ (θ), $\eta \in R$ and, by Condition (1), this takes precedence, and by the conjunction of Conditions (1) and (2.1) for $-\partial^m_{\mathrm{C}}$, we conclude $-\partial^m_{\mathrm{C}}\neg\eta$. At this stage, η is applicable and, since there are no rules for $\neg c$, we derive $+\partial_{\mathrm{C}}c$; this makes γ applicable. Now, we have an applicable meta-rule, β, for α. We have to look for meta-rules for standard rules that simply conflict with α; we have two candidates: γ and μ. There is no rule for d; hence $-\partial_{\mathrm{C}}d$ holds, and μ is discarded. On the other hand, as we just discussed, γ is applicable, thus we have to determine if it is defeated or not. Again, two options: β itself, but β is not stronger than γ; and λ which, even if it is a defeater, succeeds in defeating γ, we hence derive $+\partial^m_{\mathrm{C}}\alpha$. Finally, since α is applicable and there are no rules for $\neg b$, then $+\partial_{\mathrm{C}}b$ holds in D.

Suppose now that we extend the theory by inserting the rule ρ

$$\rho\colon (\alpha\colon a \Rightarrow_{\mathrm{C}} b) \Rightarrow_{\mathrm{O}} e$$

in R. Since ρ is in R, then it is derivable. Given that we proved $+\partial^m_{\mathrm{C}}\alpha$, ρ is applicable, we can now prove $+\partial_{\mathrm{O}}e$. This shows that it is possible to have meta-rule whose conclusion is not a rule, but a literal. Suppose now that, in addition to update R with ρ, we also replace λ with

$$\nu\colon \ldots \Rightarrow_{\mathrm{C}} \neg(\sigma\colon a \Rightarrow_{\mathrm{C}} b)$$

and include (ν, γ) in $>$. Given that σ is neither α nor ϵ, we cannot use ν to reinstate α (even if ν is stronger than γ). This means, that now $D \vdash -\partial^m_{\mathrm{C}}\alpha$, and then $D \vdash -\partial_{\mathrm{O}}e$. However, we can use ν in both Conditions (2.2) and (2.2.2.2), to prove $+\partial^m_{\mathrm{C}}\sigma$; again we have an applicable undefeated rule for b, and $+\partial_{\mathrm{C}}b$ continues to hold.

Definition 19 (Obligation Meta-Proof Conditions - Simple Variant).

$+\partial^m_{\mathrm{O}}\alpha$: *If* $P(n+1) = +\partial^m_{\mathrm{O}}\alpha$ *then*
$\exists \beta \in R^{\mathrm{O}}_{\Rightarrow}[(\alpha : A(\alpha) \hookrightarrow_{\square} c), i]$ *s.t.*
(1) β *is applicable at index* i*, and*
(2) $\forall \gamma \in R^{\mathrm{O}}[\sim(\psi : A(\alpha) \hookrightarrow_{\square} c), j] \cup R^{\mathrm{P}}[\sim(\psi : A(\alpha) \hookrightarrow_{\square} c)]$ *either*

(1) γ *is discarded (at index* j*), or*
(2) $\exists \zeta \in R^{\mathrm{O}}[(\chi : A(\alpha) \hookrightarrow_{\Box} c), k]$ *s.t.*
(1) $\chi \in \{\alpha, \psi\}$,
(2) ζ *is applicable at index* k*, and*
(3) $\zeta > \gamma$.

$-\partial^m_{\mathrm{O}}\alpha$*: If* $P(n+1) = -\partial^m_{\mathrm{O}}\alpha$ *then*
$\forall \beta \in R^{\mathrm{O}}_{\Rightarrow}[(\alpha : A(\alpha) \hookrightarrow_{\Box} c), i]$ *either*
(1) β *is discarded at index* i*, or*
(2) $\exists \gamma \in R^{\mathrm{O}}[\sim(\psi : A(\alpha) \hookrightarrow_{\Box} c), j] \cup R^{\mathrm{P}}[\sim(\psi : A(\alpha) \hookrightarrow_{\Box} c)]$*, s.t.*
(1) γ *is applicable (at index* j*), and*
(2) $\forall \zeta \in R^{\mathrm{O}}[(\chi : A(\alpha) \hookrightarrow_{\Box} c), k]$ *either*
(1) $\chi \notin \{\alpha, \psi\}$*, or*
(2) ζ *is discarded at index* k*, or*
(3) $\zeta \not> \gamma$.

The conditions to derive when a rule is proved or refuted as an obligation combine: (i) the issues to derive a literal as an obligation, and (ii) the elements illustrated above for proving a rule. Note that a deontic rule expression can appear in the body of a rule, but cannot stand on its own, thus F and R cannot contain expressions like $\mathrm{O}\alpha$.

Example 9. Let D a theory where $a, d \in F$, R contains the following rules

$\alpha\colon \ldots \Rightarrow_{\mathrm{O}} (\gamma\colon a \Rightarrow_{\mathrm{C}} c) \otimes c \otimes (\epsilon\colon d \Rightarrow_{\mathrm{O}} e \otimes f)$ $\quad$ $\beta\colon \ldots \Rightarrow_{\mathrm{C}} (\eta\colon a \Rightarrow_{\mathrm{C}} \neg c)$

$\theta\colon \mathrm{P}(\epsilon\colon d \Rightarrow_{\mathrm{O}} e \otimes f) \Rightarrow_{\mathrm{O}} \neg(\kappa\colon \mathrm{O}e, \neg e \Rightarrow_{\mathrm{O}} h)$ $\quad$ $\lambda\colon g \Rightarrow_{\mathrm{P}} (\kappa\colon \mathrm{O}, \neg e \Rightarrow_{\mathrm{O}} h)$

and $(\lambda, \theta) \in >$.

Let us consider the following derivation (where, for the sake of clarity, trivial and irrelevant steps have been removed):

1.	$+\partial^m_{\mathrm{O}}(\gamma\colon a \Rightarrow_{\mathrm{C}} c)$	α applicable for γ at index 1, $R^{\Diamond}[\neg\alpha] = \emptyset$
2.	$-\partial^m_{\mathrm{C}}(\gamma\colon a \Rightarrow_{\mathrm{C}} c)$	$\gamma \notin R$ and $R^{\mathrm{C}}[\gamma] = \emptyset$
3.	$+\partial_{\mathrm{O}} c$	1. and 2. α applicable for c at index 2, $R^{\Diamond}[\neg c] = \emptyset$
4.	$+\partial^m_{\mathrm{C}}(\eta\colon a \Rightarrow_{\mathrm{C}} \neg c)$	β applicable and $R^{\mathrm{C}}[\neg\eta] = \emptyset$
5.	$+\partial_{\mathrm{C}} a$	$a \in F$
6.	$+\partial_{\mathrm{C}} \neg c$	4. and 5. η applicable and $R^{\mathrm{C}}[\neg c] = \emptyset$
7.	$+\partial^m_{\mathrm{O}}(\epsilon\colon d \Rightarrow_{\mathrm{O}} e \otimes f)$	3. and 6. α applicable for ϵ at index 3, $R^{\Diamond}[\neg\epsilon] = \emptyset$
8.	$+\partial^m_{\mathrm{P}}(\epsilon\colon d \Rightarrow_{\mathrm{O}} e \otimes f)$	$+\partial^m_{\mathrm{O}}(\epsilon) \in P(1..7)$, Condition (1) of $+\partial^m_{\mathrm{P}}$ of Def. 20
9.	$-\partial_{\mathrm{C}} g$	$g \notin F$ and $R^{\mathrm{C}}[g] = \emptyset$
10.	$+\partial^m_{\mathrm{O}}\neg(\kappa\colon \mathrm{O}e, \neg e \Rightarrow_{\mathrm{O}} h)$	θ applicable (7.), λ discarded (9.)
11.	$+\partial_{\mathrm{C}} d$	$d \in F$
12.	$+\partial_{\mathrm{O}} e$	10. and 11. ϵ applicable for e at index 1, $R^{\Diamond}[e] = \emptyset$
13.	$-\partial_{\mathrm{C}} \neg e$	$\neg e \notin F$ and $R^{\mathrm{C}}[\neg e] = \emptyset$
14.	$-\partial_{\mathrm{O}} f$	13. ϵ discarded for f at index 2

The derivation above mostly illustrates the proof conditions for $+\partial^m_{\mathrm{O}}$ with repeated use of Condition 5 of Definition 14, and particularly the distinction between rules and literals appearing in $\otimes$-expressions: compare the combinations

Steps 1. and 2. to obtain Step 3., and Steps 3. and 6. to conclude Step 7. Note that most cases of discarded occur when there are no rules for the opposite.

Let focus on the rules in the theory. First of all, rule θ shows that it is possible to have rule expressions in the antecedent of meta-rules. Second, let us analyse the meaning of rules θ and λ: λ allows κ to be an admissible norm in the normative system. In case ϵ is present and effective (produces the effects of its conclusion, Oe), κ corresponds to a contrary-to-duty norm whose normative effect is in force in case of a violation. Typically, this kind of expression represents an accessory penalty (see [33] for the distinction between *compensatory obligations* and *contrary-to-duty obligations*). On the other hand, ϵ establishes a compensatory condition for the violation of the obligation of e. Accordingly, θ prescribes that is is forbidden to have an accessory penalty if a compensatory condition is admissible.

Definition 20 (Permission Meta-Proof Conditions - Simple Variant).

$+\partial^m_{\mathrm{P}}\alpha$: *If* $P(n+1) = +\partial^m_{\mathrm{P}}\alpha$ *then*
(1) $+\partial^m_{\mathrm{O}}\alpha \in P(1..n)$, *or*
(2) $\exists\beta \in R^{\mathrm{P}}_{\Rightarrow}[(\alpha : A(\alpha) \hookrightarrow_{\Box} c)]$ *s.t.*
(1) β *is applicable, and*
(2) $\forall\gamma \in R^{\mathrm{O}}[\sim(\varphi : A(\alpha) \hookrightarrow_{\Box} c), j]$ *either*
(1) γ *is discarded at index* j, *or*
(2) $\exists\zeta \in R^{\Diamond}[(\chi : A(\alpha) \hookrightarrow_{\Box} c), k]$ *s.t.*
(1) $\chi \in \{\alpha, \varphi\}$,
(2) ε *is applicable (at index* k*), and*
(3) $\varepsilon > \gamma$.

$-\partial^m_{\mathrm{P}}\alpha$: *If* $P(n+1) = -\partial^m_{\mathrm{P}}\alpha$ *then*
(1) $-\partial^m_{\mathrm{O}}\alpha \in P(1..n)$, *and*
(2) $\forall\beta \in R^{\mathrm{P}}_{\Rightarrow}[(\alpha : A(\alpha) \hookrightarrow_{\Box} c)]$ *either*
(1) β *is discarded, or*
(2) $\exists\gamma \in R^{\mathrm{O}}[\sim(\varphi : A(\alpha) \hookrightarrow_{\Box} c), j]$, *s.t.*
(1) γ *is applicable at index* j, *and*
(2) $\forall\zeta \in R^{\Diamond}[(\chi : A(\alpha) \hookrightarrow_{\Box} c), k]$ *either*
(1) $\chi \notin \{\alpha, \varphi\}$, *or*
(2) ε *is discarded (at index* k*), or*
(3) $\varepsilon \not> \gamma$.

Similarly to ∂^m_{O}, the conditions to determine the provability of rules as permission combine the elements of ∂_{P} and the new features included in ∂^m_{C}. As illustrated by Example 9 above, Condition (1) corresponds to the 'obligation implies permission principle' (Axiom D), characteristic of Deontic Logic.

Example 10. Consider theory D with the following four applicable rules

$$\beta\colon \ldots \Rightarrow_{\mathrm{P}} (\alpha\colon a \Rightarrow_{\mathrm{P}} b) \qquad \gamma\colon \ldots \Rightarrow_{\mathrm{C}} (\alpha\colon a \Rightarrow_{\mathrm{P}} b)$$
$$\eta\colon \ldots \Rightarrow_{\mathrm{P}} \neg(\alpha\colon a \Rightarrow_{\mathrm{P}} b) \qquad \theta\colon \ldots \Rightarrow_{\mathrm{C}} \neg(\alpha\colon a \Rightarrow_{\mathrm{P}} b).$$

It is immediate to see that β and η are in conflict with each other, and so are γ and θ (in both cases the conclusion of a rule is the complement of the other). However, in Condition (2.2) of $+\partial^m_{\mathrm{P}}$, only obligation rules attack a permissive rule, and there are no obligation rules conflicting with β (and symmetrically η); therefore, Condition (2.2) is vacuously satisfied: we have $+\partial^m_{\mathrm{P}}\alpha$ and $+\partial^m_{\mathrm{P}}\neg\alpha$. Conversely, we have to evaluate θ according to Condition (2.2.2) of $+\partial^m_{\mathrm{C}}$ to determine whether $D \vdash +\partial^m_{\mathrm{C}}\alpha$. Since θ is neither discarded nor defeated, α is not defeasibly provable, and we can repeat the argument for $\neg\alpha$. Consequently, $D \vdash -\partial^m_{\mathrm{C}}\alpha$ and $D \vdash -\partial^m_{\mathrm{C}}\neg\alpha$. The example illustrates the case where it is possible for a normative system to make admissible a norm, at that at the same time is admissible not to have that norm. However, it is not possible to have a norm, and not to have the same norm (or alternatively, to include a norm and to remove the norm at the same time).

We will use $D \vdash_S \pm\#c$ to indicate that there is a proof of $\pm\#c$ from D using the proof conditions for Simple Conflict Defeasible Deontic Logic.

5.3 Prudential Conflict Defeasible Deontic Logic

We now introduce a second variant of the meta-logic. This variant takes into account a different idea of when meta-rules are in conflict. In the previous variant, the idea was that a conflict occurs when two statements fire concurrently, one affirming that a rule holds (is, or is inserted, in the system), and the second denying the rule (i.e., asserting that the rule is not in the system, has been removed). The simple variant took into account the specific label of the rule.

As discussed in the initial part of this section, the second variant (prudential) focuses on the "content" of the rules, and is partially inspired by some statutory interpretation principles, mostly the principle of non-redundancy of the prescriptions in a statute. Suppose that we have the rules

$$\alpha\colon a \Rightarrow_{\mathrm{C}} b \qquad\qquad \beta\colon a \Rightarrow_{\mathrm{C}} \neg b.$$

The logic of Section 4 is very well-equipped to handle such a case. If a is derivable, both rules are applicable; the two rules are for opposite conclusion. The logic is sceptical, and it prevents to derive positive conclusions. It derives, that both b and $\neg b$ are refuted. Given that the two rules have exactly the same antecedent, it is *not* possible that one of them is applicable and the other is not. Therefore, (i) the combination of the two rules introduces a normative gap: when a holds, we are not able to determine if b or $\neg b$ is the case; (ii) suppose that the theory specifies that one rule is stronger than the other. In this case, we can derive the conclusion of the stronger rule. Given that the two rules have the same antecedent, and the weaker rule is always defeated, there is no situation where the weaker rule can produce its conclusion. Accordingly, the weaker rule is redundant.

The principle of non-redundancy in statutory interpretation states that every term in a statute has a purpose. Therefore, if a provision in a statute never produces an effect, then there is no purpose for such a provision, and the provision is redundant. The proof conditions we present in the rest of this section

implement two key elements: (i) at every step, they look for a rule whose content is incompatible with the content of the rule we want to prove, and (ii) they extend the superiority relation mechanism to check not only if a meta-rule is stronger than another (meta-)rule, but also the superiority for the (standard) rules themselves.

We have now all the tools to reformulate the proof tags for proving/refuting rules in Definitions 18–20 to accommodate the new ideas of conflicts introduced at the begin of Section 5.

Definition 21 (Constitutive Meta-Proof Conditions - Prudential Variant).

$+\partial^m_{\mathrm{C}}\alpha$: *If* $P(n+1) = +\partial^m_{\mathrm{C}}\alpha$ *then*
(1) $\alpha \in R$ *or*
(2) (1) $\forall \omega \in R$, ω *does not prudentially conflict with* α, *and*
(2) $\exists \beta \in R^{\mathrm{C}}_{\Rightarrow}[\alpha]$ *s.t.*
(1) β *is applicable, and*
(2) $\forall \gamma \in R^{\mathrm{C}}$ *s.t.* γ *conflicts with* β, *either*
(1) γ *is discarded, or*
(2) $\exists \zeta \in R^{\mathrm{C}}$ *s.t.* ζ *conflicts with* γ, *and*
(1) ζ *is applicable, and*
(2) (1) $\zeta > \gamma$, *or*
(2) $\gamma \not> \zeta$, *and* $C(\zeta) > C(\gamma)$.

$-\partial^m_{\mathrm{C}}\alpha$: *If* $P(n+1) = -\partial^m_{\mathrm{C}}\alpha$ *then*
(1) $\alpha \notin R$ *and*
(2) (1) $\exists \omega \in R$, ω *prudentially conflicts with* α, *or*
(2) $\forall \beta \in R^{\mathrm{C}}_{\Rightarrow}[\alpha]$ *either*
(1) β *is discarded, or*
(2) $\exists \gamma \in R^{\mathrm{C}}$ *s.t.* γ *conflicts with* β, *s.t.*
(1) γ *is applicable, or*
(2) $\forall \zeta \in R^{\mathrm{C}}$ *s.t.* ζ *conflicts with* γ, *either*
(1) ζ *is discarded, or*
(2) (1) $\zeta \not> \gamma$, *and*
(2) $\gamma > \zeta$ *or* $C(\zeta) \not> C(\gamma)$.

Notice that the only new parts wrt Definition 18 are: (i) Condition (2.1) uses prudentially conflict instead of simply conflict, and (ii) Condition (2.2.2.2.2) verifies whether the meta-rule supporting the conclusion is stronger than the meta-rule attacking the conclusion (γ); if this is not the case, we can see if the is an instance of the superiority relation for the standard rules (i.e., $C(\zeta)>C(\gamma)$).

Example 11. Consider again the theory of Example 8, with $(\lambda, \gamma) \in >$, and:

$\beta\colon \ldots \Rightarrow_{\mathrm{C}} (\alpha\colon a \Rightarrow_{\mathrm{C}} b)$ $\quad \eta\colon \ldots \Rightarrow_{\mathrm{C}} c$ $\quad \lambda\colon \ldots \rightsquigarrow_{\mathrm{C}} (\_\colon a \Rightarrow_{\mathrm{C}} b)$
$\gamma\colon c \Rightarrow_{\mathrm{C}} \neg(\epsilon\colon a \Rightarrow_{\mathrm{C}} b)$ $\quad \theta\colon \ldots \Rightarrow_{\mathrm{C}} \neg(\eta\colon \ldots \Rightarrow_{\mathrm{C}} c)$ $\quad \mu\colon d \Rightarrow_{\mathrm{C}} \neg(\alpha\colon a \Rightarrow_{\mathrm{C}} b)$.

All rules that were in conflict in the Simple Conflict variant are in conflict in the Prudential Conflict version. Now, we do not have to ensure that the label of the

standard rule that is the conclusion of λ is either the label of the rule that is the conclusion of β, or the conclusion of γ. Any rule conflicting with ϵ will do.

The following two definitions, that determine the proof conditions for obligations and permissions, are obtained directly from Definitions 19 and 20 using the changes we just explained for Definition 21.

Definition 22 (Obligation Meta-Proof Conditions - Prudential Variant).

$+\partial^m_{\mathrm{O}}\alpha$: *If* $P(n+1) = +\partial^m_{\mathrm{O}}\alpha$ *then*
$\exists\beta \in R^{\mathrm{O}}_{\Rightarrow}[\alpha, i]$ *s.t.*
(1) β *is applicable at index* i*, and*
(2) $\forall\gamma \in R^{\mathrm{O}}[\nu, j] \cup R^{\mathrm{P}}[\nu]$ *s.t.* γ *conflicts with* β*, either*
(1) γ *is discarded (at index* j*), or*
(2) $\exists\zeta \in R^{\mathrm{O}}[\chi, k]$ *s.t.* ζ *conflicts with* γ
(1) ζ *is applicable (at index* k*), and either*
(2) (1) $\zeta > \gamma$*, or*
(2) $\gamma \not> \zeta$*, and* $C(\zeta) > C(\gamma)$.

$-\partial^m_{\mathrm{O}}\alpha$: *If* $P(n+1) = -\partial^m_{\mathrm{O}}\alpha$ *then*
$\forall\beta \in R^{\mathrm{O}}_{\Rightarrow}[\alpha, i]$ *either*
(1) β *is applicable at index* i*, or*
(2) $\exists\gamma \in R^{\mathrm{O}}[\nu, j] \cup R^{\mathrm{P}}[\nu]$ *s.t.* γ *conflicts with* β*, s.t.*
(1) γ *is applicable (at index* j*), and*
(2) $\forall\zeta \in R^{\mathrm{O}}[\chi, k]$ *s.t.* ζ *conflicts with* γ *either*
(1) ζ *is discarded (at index* k*), or*
(2) (1) $\zeta \not> \gamma$*, and*
(2) $\gamma > \zeta$ *or* $C(\zeta) \not> C(\gamma)$.

Definition 23 (Permission Meta-Proof Conditions - Prudential Variant).

$+\partial^m_{\mathrm{P}}\alpha$: *If* $P(n+1) = +\partial^m_{\mathrm{P}}\alpha$ *then*
$\exists\beta \in R^{\mathrm{P}}_{\Rightarrow}[\alpha]$ *s.t.*
(1) β *is applicable, and*
(2) $\forall\gamma \in R^{\mathrm{O}}[\nu, j]$ *s.t.* γ *conflicts with* β*, either*
(1) γ *is discarded at index* j*, or*
(2) $\exists\zeta \in R^{\mathrm{O}}[\chi, k] \cup R^{\mathrm{P}}[\chi]$ *s.t.* ζ *conflicts with* γ
(1) ζ *is applicable (at index* k*), and either*
(2) (1) $\zeta > \gamma$*, or*
(2) $\gamma \not> \zeta$*, and* $C(\zeta) > C(\gamma)$.

$-\partial^m_{\mathrm{P}}\alpha$: *If* $P(n+1) = -\partial^m_{\mathrm{P}}\alpha$ *then*
$\forall\beta \in R^{\mathrm{P}}_{\Rightarrow}[\alpha]$ *either*
(1) β *is discarded, or*
(2) $\exists\gamma \in R^{\mathrm{O}}[\nu, j]$ *s.t.* γ *conflicts with* β*, s.t.*
(1) γ *is applicable at index* j*, and*

(2) $\forall \zeta \in R^{\mathrm{O}}[\chi, k] \cup R^{\mathrm{P}}[\chi]$ *s.t.* ζ *conflicts with* γ *either*
(1) ζ *is discarded (at index* k*), or*
(2) (1) $\zeta \not> \gamma$*, and*
(2) $\gamma > \zeta$ *or* $C(\zeta) \not> C(\gamma)$*.*

Example 12. Consider theory $D = (F = \{c\}, R, > = \{(\eta, \theta)\})$, with $R = \{$

$$\alpha\colon \ldots \Rightarrow_{\mathrm{C}} (\beta\colon a \Rightarrow_{\mathrm{P}} b) \qquad \gamma\colon \ldots \Rightarrow_{\mathrm{C}} (\epsilon\colon a \Rightarrow_{\mathrm{P}} \neg b)$$
$$\eta\colon c \Rightarrow_{\mathrm{P}} d \qquad \theta\colon c \Rightarrow_{\mathrm{P}} \neg d\}.$$

Rules α and γ conflict with one another, since β rationally conflicts with ϵ (and the other way around). Rules α and γ are both applicable (thus no discarded, nor stronger rule exists). Therefore, α satisfies Condition (2.2.2) of $-\partial^m_{\mathrm{C}}$ (Definition 21) to derive $-\partial^m_{\mathrm{C}}\gamma$, and γ does the same to prove $-\partial^m_{\mathrm{C}}\alpha$.

Let us focus on η and θ. These two rules are in R, thus even if the rules are in a prudential conflict (Case 2 of Definition 17), we can prove them with C modality. Their antecedent c is a fact, they are hence applicable. Are we able to prove $\mathrm{P}p$ and $\mathrm{P}\neg p$? The answer is positive. Condition (2.2) of $+\partial_{\mathrm{P}}$ (Definition 9) specifies that only obligation rules can attack a permission rule. Ergo, the two rules are individually unopposed, and we conclude $+\partial_{\mathrm{P}}p$ and $+\partial_{\mathrm{P}}\neg p$. Rules β and ϵ have the same rule structure of η and θ, and one might ask if preventing the inclusion of both rules is meaningful. Again, from the legal drafting point of view, where the aim is to include only provisions that add content that would not be drawn otherwise, a better alternative would be to adopt weak permission, where something is permitted if it is not possible to derive the obligation to the contrary. In this view, p is permitted when we prove $-\partial_{\mathrm{O}}{\sim}p$; for a detailed analysis of how to model weak permission in DDL see [35].

Note, that, even if we had an instance of the superiority relation between η and θ, it would play no role whatsoever in establishing the provability of the two permissions. However, the superiority relation is relevant, when the two permission rules are the conclusion of meta-rules. The proof conditions for the prudential variant of DDL allows us to consider two cases.

Case $(\alpha, \gamma) \in >$. This is the same circumstance adopted by the simple conflict variant. The legal intuition is that meta-rules are norms from an higher order in a legal hierarchy which provides a method to solve the conflict. Here, we conclude $+\partial^m_{\mathrm{C}}\beta$ and $-\partial^m_{\mathrm{C}}\epsilon$, and if a is derivable, we are entitled to conclude $+\partial_{\mathrm{P}}b$.

Case $(\epsilon, \beta) \in >$. Here, the normative system is equipped with a mechanism to solve the conflict between the (standard) rules, even if such a mechanism (as is the case for permission) turns out to be irrelevant for the direct solution. In this instance, we reverse the result of the case above, deriving $+\partial^m_{\mathrm{C}}\epsilon$ and $+\partial_{\mathrm{P}}\neg b$.

We shall use $D \vdash_R \pm\#c$ to indicate that there is a proof of $\pm\#c$ from D using the proof conditions for Prudential Conflict Defeasible Deontic Logic.

[Gui 6] va bene questa discussione qui?

Note that the superiority relation $>$ is defined over the union of standard rules and meta-rules; thus, it is possible to have an instance where a standard rule and a meta-rule are involved. This is useful for cases with rules like:

$$\alpha\colon a \Rightarrow_{\mathrm{O}} b \qquad \beta\colon (\gamma\colon c \Rightarrow_{\mathrm{C}} d) \Rightarrow_{\mathrm{O}} \neg b.$$

Here, it is meaningful to add a superiority relation, for instance $\alpha > \beta$, given the conclusions of the two rules are opposite. Moreover, the proof conditions for the prudential variants allow us to use instances on meta-rules and instances on rules for the derivation of rules. Also, it is possible to have opposing information about the strength of rules. Consider the following example[12]:

$$\alpha_1 \colon A(\alpha_1) \Rightarrow_C (\gamma \colon a \Rightarrow_C b) \qquad \alpha_2 \colon A(\alpha_2) \Rightarrow_C (\gamma \colon a \Rightarrow_C b)$$
$$\beta_1 \colon A(\beta_1) \Rightarrow_C (\zeta \colon a \Rightarrow_C \neg b) \qquad \beta_2 \colon A(\beta_2) \Rightarrow_C (\zeta \colon a \Rightarrow_C \neg b).$$

According to Condition 2 of Definition 17 γ and ζ prudentially conflict: they have the same antecedent and opposite conclusions. It is hence meaningful to establish that one prevails over the other, let us $\zeta > \gamma$. Given that γ and ζ conflict, the conclusions of the α rules prudentially conflict with the conclusion of the β rules (Condition 5 of Definition 17).

Therefore, again, it is meaningful to have instances of the superiority relation. Suppose we have $\alpha_1 > \beta_1$, $\beta_1 > \alpha_2$ and $\alpha_2 > \beta_2$, and that all meta-rules are body-applicable. We ask whether $+\partial^m_C \gamma$? We thus have to check the conditions set in Definition 21: since α_1 is applicable by Condition (2.2.1), we then have to corroborate Condition (2.2.2). The two β rules are applicable as well, thus we have to verify if they are defeated by stronger and applicable (meta-)rules; they are given$\alpha_1 > \beta_1$ and $\alpha_2 > \beta_2$. Accordingly, we can use Condition (2.2.2.2.1) in both cases to satisfy (2.2.2), and so conclude $+\partial^m_C \gamma$.

We now turn our attention to $+\partial^m_C \zeta$. Similarly to the other case we have an applicable rule for it (any of the two βs will do), and again there exists an applicable (meta-)rule for a rule conflicting with ζ (the two αs). We have to rebut them: (i) For α_2 - as for the case of γ above - we can apply Condition (2.2.2.2.1) and the instance of $>$ on the meta-rules $\beta_2 > \alpha_2$; (ii) For α_1 we do not have $\alpha_1 > \beta_2$, but we have $C(\beta_2) = \zeta > \gamma = C(\alpha_1)$, that which satisfies Condition (2.2.2.2.2). Hence, we conclude $+\partial^m_C \zeta$.

The intuitions of Condition (2.2.2.2) of Definition 21, as well as of Condition (2.2.2) of Definitions 22 and 23, is to provide a mechanism to solve conflicts over meta-rules. Here, if a preference over meta-rules exists, use it; otherwise, check whether we have a preference over the conflicting (standard) rule. Accordingly, instances of the superiority relation on standard rules can fill gaps on the same relation over meta-rules.

5.4 Formal Properties of the Logical Apparatus

In this subsection we introduce two derived properties to refine the definition of *extension* and the definition of *theory equivalence*. The proposed revision and incorporation permits to correctly devise such notions in the processes of checking introduced in Section 6.

Theorem 1 (Coherence). *Let D be a defeasible deontic meta-theory. There is not pair of tagged modal formulas $+\#x$ and $-\#x$ such that $D \vdash_L +\#x$ and $D \vdash_L -\#x$, for $L \in \{S, R\}$.*

[12] We are very grateful to one of the anonymous referees for pointing out the example.

Proof. The result follows from Theorem 1 of [38] proving that if the proof conditions for a positive proof tag and the corresponding negative are the strong negation of each other, then no theory can prove both $+\#x$ and $-\#x$ for any conclusion x. It is immediate to verify that all pairs of proof conditions defined in this paper obey the principle of strong negation.

Theorem 2 (Consistence). *Let D be defeasible deontic meta-theory such that the superiority relation $>$ is acyclic*[13]*. For any literal l, and rules α and φ:*

1. *If $D \vdash_L +\partial_{\mathrm{C}} l$ and $D \vdash_L +\partial_{\mathrm{C}} \neg l$, then $l, \neg l \in F$, for $L \in \{S, R\}$;*
2. *It is no possible to prove both $D \vdash_L +\partial_{\mathrm{O}} l$ and $D \vdash_L +\partial_{\mathrm{O}} l$, for $L \in \{S, R\}$.*
3. *If $D \vdash_S +\partial^m_{\Box} \alpha$ and $D \vdash_S +\partial^m_{\Box} \varphi$, such that α simply conflicts with φ, then $\alpha, \varphi \in R$;*
4. *If $D \vdash_R +\partial^m_{\Box} \alpha$ and $D \vdash_R +\partial^m_{\Box} \varphi$, such that α prudentially conflicts with φ, then $\alpha, \varphi \in R$.*

Proof. It is immediate to verify that, according to Definitions 14 and 15, no rule is applicable and discarded (for the same literal) at the same time. All proof conditions have the same structure that specifies that there is an applicable rule, and every non discarded rule for a conflicting conclusion is defeated by a stronger applicable rule. Suppose that l and $\sim l$ (α and β) are both provable for $\Box$. This means that rules $\beta \in R[l]$ and $\gamma \in R[\neg l]$, for l and $\neg l$, exist and are applicable. Given that both l and $\neg l$ are provable for $\Box$, then for every applicable rule β for l, there is an applicable rule γ' for $\neg l$ such that $\gamma' > \beta$, and the other way around. However, since the rule set is finite, the situation is possible if, only if, there is a cycle in the superiority relation (see the proof of Theorem 3.5 of [9]). A contradiction.

For the cases for C, condition (1) of the appropriate proof conditions allows us to conclude $+\partial_{\mathrm{C}} l$, $+\partial^m_{\mathrm{C}} \alpha$ when $l \in F$ and $\alpha \in R$; however, when $l \in F$ ($\alpha \in R$) it is not possible to use condition (2) to derive $+\partial_{\mathrm{C}} \neg l$ ($+\partial^m_{\mathrm{C}} \beta$), but Condition (1) applies, thus $\neg l \in F$ ($\beta \in R$). Note Cases 3. and 4. are not possible when $\Box$ is either O or P, since we do not allow deontic literal in F and there are no deontic rule expressions in R.

We can now modify the interpretation of extension of Definition 10 to adapt the notions discussed above. First, we revise the notion of Herbrand Base to include rule labels. Hence $HB(D) = \{l, \sim l \in \mathrm{PLit} | \, l \text{ appears in } D\} \cup \{\alpha, \not\alpha \in \mathrm{Lab} | \, \alpha \text{ appears in } D\}$, second we define the so called modal Herbrand Base to close the element of $HB(D)$ under the various modalities; thus $MHB(D) = \{\Box e | \, e \in HB(D) \wedge \Box \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}\}$.

Definition 24 (Extension). *Given a defeasible deontic meta-theory D, we define the* extension *of D according to variant L, $L \in \{S, R\}$, of DDL as $E_L(D) = (+\partial_{\Box}, -\partial_{\Box}, +\partial^m_{\Box}, -\partial^m_{\Box})$, where $\pm\partial_{\Box} = \{l \in \mathrm{PLit} | \, l \in HB(D) \wedge D \vdash_L \pm\partial_{\Box} l\}$ and $\pm\partial_{\Box} = \{\alpha \in \mathrm{Lab} | \, \alpha \in HB(D) \wedge D \vdash_L \pm\partial^m_{\Box} l\}$, with $\Box \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$.*

[13] A relation is acyclic if the transitive closure of it does not contain a cycle.

We say that two theories are equivalent when they conclude the same conclusions, that is they have the same extension. Formally:

Definition 25 (Theory Equivalence). *Two defeasible, deontic meta-theories D and D' are* equivalent*, notationally $D \equiv D'$, iff $E_L(D) \equiv E_L(D')$.*

The notion of theory equivalence will play a key role in proving that our algorithms are sound and complete.

6 Algorithms

The algorithms presented in this work compute the extension of a defeasible deontic meta-theory; they take inspiration from previous versions [35,66,34].

The main idea of the algorithms being to compute, at each iteration step, a *simpler* yet equivalent theory than the one we had at the previous step. By 'simpler theory', we mean that, by proving and disproving/rejecting literals and standard rules, we can progressively do two actions:

1. **Simplify the antecedents** of the rules of the theory, by removing (previously) proved elements;
2. **Eliminate (discarded) rules** from the theory itself.

To better understand how such actions operate and, most importantly, why they are possible, we firstly want the reader to focus on some key concepts of Defeasible Logic and its implementations.

The first concept is that, being our logical apparatus non-monotonic, when we want to prove something (either a statement or a rule), we cannot consider *a single rule* supporting such a conclusion; we need to consider, at the same time, (i) All the rules supporting it, as well as (ii) All the rules supporting the opposite conclusion, as such rules must be "defeated" in order to prove. (Naturally, if the "opposing" teams prevails, we shall prove the opposite conclusion, whereas if neither team wins, we refute both.)

Do we really have to defeat all the supporting/opposing rules to prove our conclusion? The answer to that question is: No, we do not consider the discarded rules, but only the applicable ones.

It is natural then to ask what makes a rule applicable/discarded. The rule set (and the superiority relation establishing the relative strength of rules) should be as general as possible, so to be able to describe as many situations as possible.

On the other hand, the set of facts is used to describe the current, specific situation. Imagine, for instance, all the rules describing the normative corpus of driving. There are rules that impose to fasten the seat belts while driving, others imposing to wear an helmet while riding a motorbike. What settles if we need to fasten my seat belt, or if we have to wear an helmet is the set of facts. Once the set of facts is fed as input, of all the rules present in the initial rule sets, some will be applicable (and they can hence contribute in the team fight), whereas some are discarded (and thus it is like they were not even present in the initial rule sets given that specific set of facts).

When we move from standard to meta-rules, two considerations are in order.

First, the set of rules supporting a given literal may not be (entirely) determined at the start of the computation. Standard rules can be conclusions of meta-rules, to determine the derivability/refutability of such a literal (and its opposite), we need to wait until all the supporting and opposing standard rules that are conclusion of meta-rules, have been either proved, or refuted.

Second, a rule being discarded is different from a rule being not provable. Therefore, we cannot discard a meta-rule because its antecedent contains a discarded standard rule: the standard rule can be discarded, but still be provable.

Our algorithms make good use of these concepts: as the input is a theory with a *specific* set of facts (thus describing a *specific situation*), they reduce, step after step, the number of rules of the theory itself by eliminating, at iteration $i+1$, all those rules that have, in their antecedent, an element that has been *refuted* at iteration i (as such rules are discarded by Definition 15).

Symmetrically, if an element has been proved at iteration i, at iteration $i+1$ we can safely remove it from the antecedents of all the rules it appears in, as their applicability does not depend any longer from such an element.

Roughly said, a rule is applicable when everything in its antecedent has been proved. Consequently, a rule with empty antecedent is always vacuously applicable, as there is nothing (more) to prove.

A rule is discarded if (at least) one of its antecedent's elements has been previously rejected. When a rule is discarded, it can no longer play part in neither supporting its conclusion, nor rejecting the opposite. Accordingly (as it shall be proved afterwards), theories with or without a discarded rule have the same extension.

Consider $D = (F, R, > = \{(\beta, \theta), (\theta, \zeta), (\beta, \psi)\})$ be a theory such that

$$R = \{\alpha\colon \ldots \Rightarrow_{\mathrm{C}} z \qquad \beta\colon b, z \Rightarrow_{\mathrm{C}} \neg l \qquad \zeta\colon \ldots \Rightarrow_{\mathrm{C}} \neg l$$
$$\theta\colon \neg z \Rightarrow_{\mathrm{C}} \neg l \qquad \psi\colon \ldots \Rightarrow_{\mathrm{C}} l\},$$

and assume that, at iteration i, the algorithm proves $+\partial_{\mathrm{C}} z$: at the same at iteration i, the algorithm proves $-\partial_{\mathrm{C}} \neg z$ as well (Procedure ProveLiteral, Lines 8 and 9). At iteration $i+1$, we can hence perform two operations:

1. Remove z from β's antecedent: β's applicability no longer depends upon z, but solely on b, the only remaining element in its antecedent; the "new β" will be
$$\beta\colon b \Rightarrow_{\mathrm{C}} \neg l.$$
2. Eliminate θ from R: θ is discarded given that $\neg z \in A(\theta)$; therefore,
$$R^{i+1} = R^i \setminus \{\theta\}.$$

Naturally, once θ has been eliminated from the rule set, we can adjust accordingly the superiority relation by removing all those tuples that have θ as element – in this case, (θ, α) and (β, θ). The only "potentially" supporting rule for l is now ψ, which is "potentially" defeated by β ("potentially", as we need to establish whether β and ψ are applicable).

Description of Algorithm 1 Deontic Meta Extension

The main Algorithm 1 starts by initialising

- The ∂ sets for the extension (Line 1).
- The Modal Herbrand Base *MHB*, used to properly cycle over literals (Lines 19–29), and over standard rules (Lines 32–42) of the theory.
- Support sets $R^{\Box}[]$ (Line 4), one for each literal in the modal Herbrand Base; given a modal literal Xl, $R^{X}[l]$ plays a fundamental role in proving/refuting it, as some standard rules (α in the algorithm) supporting Xl may not be in the initial rule set, but be the conclusion of meta-rules (β in the algorithm). The decidability of Xl (and naturally that of $\neg Xl$) must thus be postponed until we have proved/rejected such 'α rules'.
- Support sets $R[]^{\Box}_{infd}$ (Line 5), one for each literal in *MHB*, and used during the "team fight"; given a modal literal Xl, $R[l]^{X}_{infd}$ stores the rules supporting l that which are defeated by an *applicable* rule for $\sim l$ (for a more detailed explanation, see below during description of Procedure CheckLiteral).

Algorithm 1: Deontic Meta Extension

Input: Defeasible deontic meta-theory D, $variant = \{\text{simply, rationally}\}$
Output: The meta-extension $E_L(D)$

$\pm\partial_{\Box} \leftarrow \emptyset$; $\pm\partial^{meta}_{\Box} \leftarrow \emptyset$;
modalHerbrandBase(MHB)
for $\Box l \in MHB$ **do**
  $R^{\Box}[l] \leftarrow \{\alpha$ appears in $R | \, \alpha \in R^{\Box}[l] \vee (\exists \beta \in R^{\blacksquare}[\alpha] \wedge l \in C(\alpha)$ s.t. $\alpha \colon A(\alpha) \hookrightarrow_{\Box} C(\alpha))\}$;
  $R[l]^{\Box}_{infd} \leftarrow \emptyset$;
end
Conflicts();
for $\alpha \in R^{\mathrm{O}}$ **do**
  $k =$ `length`$(C(\alpha))$, Build matrix $\alpha[2][k]$;
  for $i = 1$ *to* 2, $j = 1$ *to* k **do** Set $\alpha[i][j]$ to *null*;
end
for $l \in F$ **do** Prove(l, C); Refute$(\sim l, \mathrm{C})$;
for $\alpha \in R$ **do**
  Prove(α, C);
  Refute(γ, C), s.t. $\gamma \notin R$ and α *variant* conflicts with γ;
end
repeat
  $\partial^{\pm}_{\Box} \leftarrow \emptyset$;
  for $\Box l \in MHB$ **do**
    if $R^{\Box}[l] = \emptyset$ **then** Refute$(l, \Box)$;
    switch *modality* $\Box$ **do**
      case $\Box \in \{\mathrm{C}, \mathrm{P}\}$ **do**
        if $\exists \beta \in R^{\Box}[l]. \, A(\beta) = \emptyset \wedge \beta \in +\partial^{meta}_{\Box}$ **then** CheckLiteral$(l, \Box, \beta)$;
      end
      case $\Box = \mathrm{O}$ **do**
        if $\exists \beta \in R^{\mathrm{O}}[l, i]. \, A(\beta) = \emptyset \wedge \forall j < i. \, \beta[1][j] = +$ *and* $\beta[2][j] = + \wedge \beta \in +\partial^{meta}_{\Box}$ **then** CheckLiteral(l, O, β);
      end
    end
  end
  $\pm\partial_{\Box} \leftarrow \pm\partial_{\Box} \cup \partial^{\pm}_{\Box}$;
  $\partial^{\pm}_{meta,\Box} \leftarrow \emptyset$;
  for $\Box\alpha \in MHB$ **do**
    if $R^{\Box}[\alpha] = \emptyset$ **then** Refute$(\alpha, \Box)$;
    switch *modality* $\Box$ **do**
      case $\Box \in \{\mathrm{C}, \mathrm{P}\}$ **do**
        if $\exists \beta \in R^{\Box}[\alpha]. \, A(\beta) = \emptyset$ **then** CheckRule$(\alpha, \Box, \beta)$;
      end
      case $\Box = \mathrm{O}$ **do**
        if $\exists \beta \in R^{\mathrm{O}}[\alpha, i]. \, A(\beta) = \emptyset \wedge \forall j < i. \, \beta[1][j] = + \wedge \beta[2][j] = +$ **then** CheckRule$(\alpha, \mathrm{O}, \beta)$;
      end
    end
  end
  $\pm\partial^{meta}_{\Box} \leftarrow \pm\partial^{meta}_{\Box} \cup \partial^{\pm}_{meta,\Box}$;
until $\partial^{+}_{\Box} = \emptyset \wedge \partial^{-}_{\Box} = \emptyset \wedge \partial^{+}_{meta,\Box} = \emptyset \wedge \partial^{-}_{meta,\Box} = \emptyset$;
return $E_L(D) = \big(+\partial_{\Box}, -\partial_{\Box}, +\partial^{meta}_{\Box}, -\partial^{meta}_{\Box}\big)$

– Support sets $R[\alpha]^{\Box}_{opp}$ and $R[\alpha]^{\Box}_{supp}$, by invoking Procedure Conflicts. Such sets are determined according to which *variant* of conflict (simple or rational) is given as input, and they perfectly match Condition 1 (resp. Conditions 1–5) of Definition 16 (resp. Definition 17 if) if *variant* = *simply* (resp. *variant* = *rationally*). We decided to graphically report in full such sets for *variant* = *simply* (Lines 7–8, 11–12, and 15–16) only to point out the signatures of the rules involved: $\sim\varphi$ has same antecedent, modality, and conclusion of α, whereas χ is either α, or φ.

Procedure Conflicts

Input:

1 **for** $\big(\alpha\colon A(\alpha) \hookrightarrow_{\Box} C(\alpha)\big)$ *s.t.* α *appears in* R **do**
2 $\quad R^{\blacksquare}[\alpha] \leftarrow \{\beta \in R^{\blacksquare} \mid \alpha \in C(\beta)\}$;
3 $\quad R[\alpha]^{\blacksquare}_{infd} \leftarrow \emptyset$;
4 $\quad$ **if** *variant* = *simply* **then** `// Simple conflicts`
5 $\quad\quad$ **switch** *modality* X **do**
6 $\quad\quad\quad$ **case** $X = \mathrm{C}$ **do**
7 $\quad\quad\quad\quad R[\alpha]^{\mathrm{C}}_{opp} \leftarrow \{\gamma \in R^{\mathrm{C}}[\sim\varphi] \mid \sim\big(\varphi : A(\alpha) \hookrightarrow_{\Box} C(\alpha)\big)\}$;
8 $\quad\quad\quad\quad R[\alpha]^{\mathrm{C}}_{supp} \leftarrow \big\{\zeta \in R^{\mathrm{C}}[\chi] \mid \exists\gamma \in R[\alpha]^{\mathrm{C}}_{opp} \cap R[\sim\varphi].\, \chi \in \{\alpha, \varphi\}\big\}$;
9 $\quad\quad\quad$ **end**
10 $\quad\quad\quad$ **case** $X = \mathrm{O}$ **do**
11 $\quad\quad\quad\quad R[\alpha]^{\mathrm{O}}_{opp} \leftarrow \{\gamma \in R^{\Diamond}[\sim\varphi] \mid \sim\big(\varphi : A(\alpha) \hookrightarrow_{\Box} C(\alpha)\big)\}$;
12 $\quad\quad\quad\quad R[\alpha]^{\mathrm{O}}_{supp} \leftarrow \big\{\zeta \in R^{\mathrm{O}}[\chi] \mid \exists\gamma \in R[\alpha]^{\mathrm{O}}_{opp} \cap R[\sim\varphi].\, \chi \in \{\alpha, \varphi\}\big\}$;
13 $\quad\quad\quad$ **end**
14 $\quad\quad\quad$ **case** $X = \mathrm{P}$ **do**
15 $\quad\quad\quad\quad R[\alpha]^{\mathrm{P}}_{opp} \leftarrow \{\gamma \in R^{\mathrm{O}}[\sim\varphi] \mid \sim\big(\varphi : A(\alpha) \hookrightarrow_{\Box} C(\alpha)\big)\}$;
16 $\quad\quad\quad\quad R[\alpha]^{\mathrm{P}}_{supp} \leftarrow \big\{\zeta \in R^{\Diamond}[\chi] \mid \exists\gamma \in R[\alpha]^{\mathrm{P}}_{opp} \cap R[\sim\varphi].\, \chi \in \{\alpha, \varphi\}\big\}$;
17 $\quad\quad\quad$ **end**
18 $\quad\quad$ **end**
19 $\quad$ **else** `// Rational conflicts`
20 $\quad\quad$ **switch** *modality* X **do**
21 $\quad\quad\quad$ **case** $X = \mathrm{C}$ **do**
22 $\quad\quad\quad\quad R[\alpha]^{\mathrm{C}}_{opp} \leftarrow \{\gamma \in R^{\mathrm{C}}[\varphi] \mid \varphi$ rationally conflicts with $\alpha\}$;
23 $\quad\quad\quad\quad R[\alpha]^{\mathrm{C}}_{supp} \leftarrow \{\zeta \in R^{\mathrm{C}}[\chi] \mid \exists\gamma \in R[\alpha]^{\mathrm{C}}_{opp} \cap R^{\mathrm{C}}[\varphi].\, \chi$ conflicts with $\varphi\}$;
24 $\quad\quad\quad$ **end**
25 $\quad\quad\quad$ **case** $X = \mathrm{O}$ **do**
26 $\quad\quad\quad\quad R[\alpha]^{\mathrm{O}}_{opp} \leftarrow \{\gamma \in R^{\Diamond}[\varphi] \mid \varphi$ rationally conflicts with $\alpha\}$;
27 $\quad\quad\quad\quad R[\alpha]^{\mathrm{O}}_{supp} \leftarrow \{\zeta \in R^{\mathrm{O}}[\chi] \mid \exists\gamma \in R[\alpha]^{\mathrm{O}}_{opp} \cap R^{\Diamond}[\varphi].\, \chi$ rationally conflicts with $\varphi\}$;
28 $\quad\quad\quad$ **end**
29 $\quad\quad\quad$ **case** $X = \mathrm{P}$ **do**
30 $\quad\quad\quad\quad R[\alpha]^{\mathrm{P}}_{opp} \leftarrow \{\gamma \in R^{\mathrm{O}}[\varphi] \mid \varphi$ rationally conflicts with $\alpha\}$;
31 $\quad\quad\quad\quad R[\alpha]^{\mathrm{P}}_{supp} \leftarrow \{\zeta \in R^{\Diamond}[\chi] \mid \exists\gamma \in R[\alpha]^{\mathrm{P}}_{opp} \cap R^{\mathrm{O}}[\varphi].\, \chi$ rationally conflicts with $\varphi\}$;
32 $\quad\quad\quad$ **end**
33 $\quad\quad$ **end**
34 $\quad$ **end**
35 **end**

– Support matrices $\alpha[2][k]$[14] for every obligation rule (**for** cycle at Line 8–11), where k is the number of elements in the $\otimes$-chain. Such matrices are to verify the applicability of an obligation rule at a given index j of its $\otimes$-chain according to Condition 4 of Definition 14.

Cycle **for** of Line 12 proves (as constitutive[15]) all literals in the initial set of facts, and rejects all opposites. Symmetrically, every rule in the initial rule set (thus not standard rules that can be derived through meta-rules) are proved as constitutive rules (Line 14), and we refute standard rules that (i) are conclusions of meta-rules, and (ii) conflict with a given rule (Line 15).

The algorithm now enters the main cycle (**repeat**–**until**, Lines 17–44); cycle **for** at Lines 19–29 runs over literals, the one at Lines 32–42 runs over rules: they behave in an identical way. Naturally, if there is no rule supporting an element in the Modal Herbrand Base, we refute it (**if** at Line 20 for literals, **if** at Line 33 for rules). On the other hand, if there exists an applicable rule β that has been proved, we invoke CheckLiteral, or CheckRule, according to whether the element is a literal or a rule (see below the detailed description on how they operate) to verify if, at the current iteration, we can already prove the element at hand; note that the **if** check at Line 26 is specific for proving an element as obligatory, and differs from the one at Line 23 (same for Line 39 with respect to Line 36) because we must consider $\otimes$-chains, hence satisfy Condition 4 of Definition 14.

The algorithm terminates when no modifications on the extension are made. We remind the reader that we still use the convention that $\Box$ and $\blacksquare$ represent all three modalities whilst $\Diamond$ is restricted to obligations and permissions only, thus $\Box = \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$ and $\Diamond = \{\mathrm{O}, \mathrm{P}\}$.

Description of Procedure CheckLiteral.

Once an *applicable* and proved rule β for l has been found, we can update the set of the opposite and *defeated* rules $R[\sim l]_{infd}$ with all the γ rules that are defeated by β. We remind the reader that

– Only constitutive rules oppose constitutive to rules ($X = \mathrm{C}$, Line 3);
– Both obligation and permission rules oppose obligation rules ($X = \mathrm{O}$, Lines 13 and 14);
– Only obligation rules oppose permissive rules ($X = \mathrm{P}$, Line 24).

If there are no opposite rules stronger than β, we can refute the contrary conclusion (Lines 5, 16, and 26).

Moreover, if: (i) all the opposite rules are defeated ($R[\sim l] \setminus R[\sim l]_{infd} = \emptyset$), (ii) and at least one of the supporting, applicable rules (for Xl) is a defeasible rule (thus, not a defeater), then we can prove Xl (and we do not even care whether the remaining rules in $R[\sim l]$ are applicable or discarded – Lines 7, 18, and 28). Note that: (i) when we prove a literal as obligatory, (i.a) we prove it as permitted as well and (i.b) we refute the opposite as both (Lines 19 and 20), and (ii) when

[14] We chose indices ranging 1 to n, instead of typical implementations, for the sake of readability.)

[15] Note that facts are constitutive statements, i.e., plain literal.

we prove a literal as permitted, we refute the opposite as obligatory but not as permitted (if it is permitted to smoke, "not smoking" cannot be mandatory but it can be permitted – Lines 29 and 30).

Procedure CheckLiteral(l, X, β)

Input: A plain literal $l \in \mathrm{PLit}$, a modality $X \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$, a rule β

1 **switch** *modality* X **do**
2 **case** $X = \mathrm{C}$ **do**
3 $R[\sim l]^{\mathrm{C}}_{infd} \leftarrow R[\sim l]^{\mathrm{C}}_{infd} \cup \{\gamma \in R^{\mathrm{C}}[\sim l] \mid \beta > \gamma\}$;
4 **if** $\{\gamma \in R^{\mathrm{C}}[\sim l] \mid \gamma > \beta\} = \emptyset$ **then**
5 RefuteLiteral$(\sim l, \mathrm{C})$;
6 **end**
7 **if** $\left(R^{\mathrm{C}}[\sim l] \setminus R[\sim l]^{\mathrm{C}}_{infd} = \emptyset\right) \wedge \left(\exists \zeta \in R^{\mathrm{C}}_{\mathrm{d}}[l].\, A(\zeta) = \emptyset\right)$ **then**
8 ProveLiteral(l, C);
9 RefuteLiteral$(\sim l, \mathrm{C})$;
10 **end**
11 **end**
12 **case** $X = \mathrm{O}$ **do**
13 $R[\sim l]^{\mathrm{O}}_{infd} \leftarrow R[\sim l]^{\mathrm{O}}_{infd} \cup \{\gamma \in R^{\diamond}[\sim l] \mid \beta > \gamma\}$;
14 $R[\sim l]^{\mathrm{P}}_{infd} \leftarrow R[\sim l]^{\mathrm{P}}_{infd} \cup \{\gamma \in R^{\diamond}[\sim l] \mid \beta > \gamma\}$;
15 **if** $\{\gamma \in (R^{\mathrm{O}}[\sim l] \cup R^{\mathrm{P}}[\sim l]) \mid \gamma > \beta\} = \emptyset$ **then**
16 RefuteLiteral$(\sim l, \mathrm{O})$;
17 **end**
18 **if** $\left((R^{\mathrm{O}}[\sim l] \cup R^{\mathrm{P}}[\sim l]) \setminus R[\sim l]^{\mathrm{O}}_{infd} = \emptyset\right) \wedge \left(\exists \zeta \in R^{\mathrm{O}}_{\mathrm{d}}[l].\, A(\zeta) = \emptyset\right)$ **then**
19 ProveLiteral(l, O); RefuteLiteral$(\sim l, \mathrm{O})$;
20 ProveLiteral(l, P); RefuteLiteral$(\sim l, \mathrm{P})$;
21 **end**
22 **end**
23 **case** $X = \mathrm{P}$ **do**
24 $R[\sim l]^{\mathrm{P}}_{infd} \leftarrow R[\sim l]^{\mathrm{P}}_{infd} \cup \{\gamma \in R^{\mathrm{O}}[\sim l] \mid \beta > \gamma\}$;
25 **if** $\{\gamma \in R^{\mathrm{O}}[\sim l] \mid \gamma > \beta\} = \emptyset$ **then**
26 RefuteLiteral$(\sim l, \mathrm{O})$;
27 **end**
28 **if** $\left(R^{\mathrm{O}}[\sim l] \setminus R[\sim l]^{\mathrm{P}}_{infd} = \emptyset\right) \wedge \left(\exists \zeta \in R^{\mathrm{P}}_{\mathrm{d}}[l].\, A(\zeta) = \emptyset\right)$ **then**
29 ProveLiteral(l, P);
30 RefuteLiteral$(\sim l, \mathrm{O})$;
31 **end**
32 **end**
33 **end**

Description of Procedures ProveLiteral and RefuteLiteral

As described at the start of this section, every time we prove a literal, we can remove it from each antecedent it appears in; as negative deontic literals (e.g., $\neg \mathrm{O} {\sim} l$ which is satisfied when $+\partial_{\mathrm{O}} l$ or $+\partial_{\mathrm{P}} l$) may be in the antecedents as well, Procedure ProveLiteral takes care of such situations at Lines 16, 19, 27, and 30.

Procedure ProveLiteral(l, X)

Input: A literal $l \in \mathrm{PLit}$, a modality $X \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$
1 $\partial^+_X \leftarrow \partial^+_X \cup \{l\}$;
2 $MHB \leftarrow MHB \setminus \{Xl\}$;
3 **switch** *modality* X **do**
4 **case** $X = \mathrm{C}$ **do**
5 **for** $\Box p \in MHB$ **do**
6 $R^{\Box}[p] \leftarrow \{A(\beta) \setminus \{l\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[p]\}$;
7 **end**
8 **for** $\Box \alpha \in MHB$ **do**
9 $R^{\Box}[\alpha] \leftarrow \{A(\beta) \setminus \{l\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[\alpha]\}$;
10 **end**
11 **for** $\beta \in R^{\mathrm{O}}[l, i]$ **do** $\beta[2][i] \leftarrow -$;
12 **for** $\gamma \in R^{\mathrm{O}}[{\sim} l, j]$ **do** $\gamma[2][j] \leftarrow +$;
13 **end**
14 **case** $X = \mathrm{O}$ **do**
15 **for** $\Box p \in MHB$ **do**
16 $R^{\Box}[p] \leftarrow \{A(\beta) \setminus \{\mathrm{O}l, \neg \mathrm{O} {\sim} l, \neg \mathrm{P} {\sim} l\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[p]\} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg \mathrm{O}l\} \in A(\beta)\}$;
17 **end**
18 **for** $\Box \alpha \in MHB$ **do**
19 $R^{\Box}[\alpha] \leftarrow \{A(\beta) \setminus \{\mathrm{O}l, \neg \mathrm{O} {\sim} l, \neg \mathrm{P} {\sim} l\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[\alpha]\} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg \mathrm{O}l\} \in A(\beta)\}$;
20 $R[\alpha]^{\Box}_{opp} \leftarrow R[\alpha]^{\Box}_{opp} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg \mathrm{O}l\} \in A(\beta)\}$;
21 $R[\alpha]^{\Box}_{supp} \leftarrow R[\alpha]^{\Box}_{supp} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg \mathrm{O}l\} \in A(\beta)\}$;
22 **end**
23 **for** $\beta \in R^{\mathrm{O}}[l, i]$ **do** $\beta[1][i] \leftarrow +$;
24 **end**
25 **case** $X = \mathrm{P}$ **do**
26 **for** $\Box p \in MHB$ **do**
27 $R^{\Box}[p] \leftarrow \{A(\beta) \setminus \{\mathrm{P}l, \neg \mathrm{O} {\sim} l\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[p]\} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg \mathrm{P}l\} \in A(\beta)\}$;
28 **end**
29 **for** $\Box \alpha \in MHB$ **do**
30 $R^{\Box}[\alpha] \leftarrow \{A(\beta) \setminus \{\mathrm{P}l, \neg \mathrm{O} {\sim} l\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[\alpha]\} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg \mathrm{P}l\} \in A(\beta)\}$;
31 $R[\alpha]^{\Box}_{opp} \leftarrow R[\alpha]^{\Box}_{opp} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg \mathrm{P}l\} \in A(\beta)\}$;
32 $R[\alpha]^{\Box}_{supp} \leftarrow R[\alpha]^{\Box}_{supp} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg \mathrm{P}l\} \in A(\beta)\}$;
33 **end**
34 **end**
35 **end**

Orthogonally, every time we refute a literal, Procedure RefuteLiteral removes all those rules that have such a literal in their antecedent.

Finally, we update the support matrices $\alpha[][]$, according to Condition 4.a of Definitions 14 and 15 at Lines 11, 12, and 23 of Procedure ProveLiteral, and Line 23 of Procedure RefuteLiteral.

Procedure RefuteLiteral(l,X)

Input: A literal $l \in \mathrm{PLit}$, a modality $X \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$

1 $\partial_X^- \leftarrow \partial_X^- \cup \{l\}$;
2 $MHB \leftarrow MHB \setminus \{Xl\}$;
3 **if** $X = \mathrm{C}$ **then**
4 | **for** $\Box p \in MHB$ **do** $R^{\Box}[p] \leftarrow R^{\Box}[p] \setminus \{\beta \in R^{\Box}[p] | \, l \in A(\beta)\}$;
5 | **for** $\Box\alpha \in MHB$ **do**
6 | | $R^{\Box}[\alpha] \leftarrow R^{\Box}[\alpha] \setminus \{\beta \in R^{\Box}[\alpha] | \, l \in A(\beta)\}$;
7 | | $R[\alpha]^{\Box}_{opp} \leftarrow R[\alpha]^{\Box}_{opp} \setminus \{\beta \in R[\alpha]^{\Box}_{opp} | \, l \in A(\beta)\}$;
8 | | $R[\alpha]^{\Box}_{supp} \leftarrow R[\alpha]^{\Box}_{opp} \setminus \{\beta \in R[\alpha]^{\Box}_{supp} | \, l \in A(\beta)\}$
9 | **end**
10 | $><\!\!-\!\!> \setminus \{(\beta,\gamma),(\gamma,\beta) \in >\, | \, l \in A(\beta)\}$;
11 **else**
12 | **for** $\Box p \in MHB$ **do**
13 | | $R^{\Box}[p] \leftarrow \{A(\beta) \setminus \{\neg Xl\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[p]\} \setminus \{\beta \in R^{\Box}[p] | \, Xl \in A(\beta)\}$;
14 | **end**
15 | **for** $\Box\alpha \in MHB$ **do**
16 | | $R^{\Box}[\alpha] \leftarrow \{A(\beta) \setminus \{\neg Xl\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[\alpha]\} \setminus \{\beta \in R^{\Box}[\alpha] | \, Xl \in A(\beta)\}$;
17 | | $R[\alpha]^{\Box}_{opp} \leftarrow R[\alpha]^{\Box}_{opp} \setminus \{\beta \in R[\alpha]^{\Box}_{opp} | \, Xl \in A(\beta)\}$;
18 | | $R[\alpha]^{\Box}_{supp} \leftarrow R[\alpha]^{\Box}_{supp} \setminus \{\beta \in R[\alpha]^{\Box}_{supp} | \, Xl \in A(\beta)\}$
19 | **end**
20 | $><\!\!-\!\!> \setminus \{(\beta,\gamma),(\gamma,\beta) \in >\, | \, Xl \in A(\beta)\}$;
21 **end**
22 **if** $X = \mathrm{O}$ **then**
23 | **for** $\beta \in R^{\mathrm{O}}[l,i]$ **do** $\beta[1][i] \leftarrow -$;
24 **end**

Description of Procedures CheckRule, ProveRule and RefuteRule

We shall discuss only key distinctions with respects to the procedures for literals, as the sequence of operations and the basic ideas of Procedures CheckRule, ProveRule and RefuteRule are fundamentally equivalent to each other.

Procedure CheckRule(α, X, β)

Input: A rule α, a modality X, a rule β

switch *modality* X **do**
  case $X = \mathrm{C}$ **do**
    if *variant* = *simply* **then**
      $R[\alpha]^{\mathrm{C}}_{infd} \leftarrow R[\alpha]^{\mathrm{C}}_{infd} \cup \{\gamma \in R[\alpha]^{\mathrm{C}}_{opp} \mid \beta$ *simply* conflicts with $\gamma \wedge \beta > \gamma\}$;
    else
      $R[\alpha]^{\mathrm{C}}_{infd} \leftarrow R[\alpha]^{\mathrm{C}}_{infd} \cup \{\gamma \in R[\alpha]^{\mathrm{C}}_{opp} \cap R^{\mathrm{C}}[\varphi] \mid \beta$ *rationally* conflicts with $\gamma \wedge (\beta > \gamma) \vee (\gamma \not> \beta \wedge \alpha > \varphi)\}$;
    end
    if $\left(R[\alpha]^{\mathrm{C}}_{opp} \setminus R[\alpha]^{\mathrm{C}}_{infd} = \emptyset\right) \wedge \left(\exists \chi \in R^{\mathrm{C}}_{\mathrm{d}}[\alpha] \cap R[\alpha]^{\mathrm{C}}_{supp}.\, A(\chi) = \emptyset\right)$ **then**
      ProveRule(α, C);
      for $\varphi \in C(\gamma).\, \gamma \in R[\alpha]^{\mathrm{C}}_{opp} \wedge \varphi$ *variant conflicts with* α **do**
        RefuteRule(φ, C);
      end
    end
  end
  case $X = \mathrm{O}$ **do**
    if *variant* = *simply* **then**
      $R[\alpha]^{\mathrm{O}}_{infd} \leftarrow R[\alpha]^{\mathrm{O}}_{infd} \cup \{\gamma \in R[\alpha]^{\mathrm{O}}_{opp} \mid \beta$ *simply* conflicts with $\gamma \wedge \beta > \gamma\}$;
    else
      $R[\alpha]^{\mathrm{O}}_{infd} \leftarrow R[\alpha]^{\mathrm{C}}_{infd} \cup \{\gamma \in R[\alpha]^{\mathrm{O}}_{opp} \cap R^{\mathrm{O}}[\varphi] \mid \beta$ *rationally* conflicts with $\gamma \wedge (\beta > \gamma) \vee (\gamma \not> \beta \wedge \alpha > \varphi)\}$;
    end
    if *variant* = *simply* **then**
      $R[\alpha]^{\mathrm{P}}_{infd} \leftarrow R[\alpha]^{\mathrm{P}}_{infd} \cup \{\gamma \in R[\alpha]^{\mathrm{O}}_{opp} \mid \beta$ *simply* conflicts with $\gamma \wedge \beta > \gamma\}$;
    else
      $R[\alpha]^{\mathrm{P}}_{infd} \leftarrow R[\alpha]^{\mathrm{P}}_{infd} \cup \{\gamma \in R[\alpha]^{\mathrm{O}}_{opp} \cap R^{\mathrm{O}}[\varphi] \mid \beta$ *rationally* conflicts with $\gamma \wedge (\beta > \gamma) \vee (\gamma \not> \beta \wedge \alpha > \varphi)\}$;
    end
    if $\left(R[\alpha]^{\mathrm{O}}_{opp} \setminus R[\alpha]^{\mathrm{O}}_{infd} = \emptyset\right) \wedge \left(\exists \chi \in R^{\mathrm{O}}_{\mathrm{d}}[\alpha].\, A(\chi) = \emptyset\right)$ **then**
      ProveRule(α, O); ProveRule(α, P);
      for $\gamma \in R[\alpha]^{\mathrm{O}}_{opp}.\, \varphi \in C(\gamma) \wedge \varphi$ *variant* conflicts with α **do**
        RefuteRule(φ, O); RefuteRule(φ, P);
      end
    end
  end
  case $X = \mathrm{P}$ **do**
    if *variant* = *simply* **then**
      $R[\alpha]^{\mathrm{P}}_{infd} \leftarrow R[\alpha]^{\mathrm{P}}_{infd} \cup \{\gamma \in R[\alpha]^{\mathrm{P}}_{opp} \mid \beta$ *simply* conflicts with $\gamma \wedge \beta > \gamma\}$;
    else
      $R[\alpha]^{\mathrm{P}}_{infd} \leftarrow R[\alpha]^{\mathrm{P}}_{infd} \cup \{\gamma \in R[\alpha]^{\mathrm{P}}_{opp} \cap R^{\mathrm{P}}[\varphi] \mid \beta$ *rationally* conflicts with $\gamma \wedge (\beta > \gamma) \vee (\gamma \not> \beta \wedge \alpha > \varphi)\}$;
    end
    if $R[\alpha]^{\mathrm{P}}_{opp} \setminus R[\alpha]^{\mathrm{P}}_{infd} = \emptyset \wedge \exists \chi \in R^{\mathrm{P}}_{\mathrm{d}}[\alpha].\, A(\chi) = \emptyset$ **then**
      ProveRule(α, P);
      for $\gamma \in R[\alpha]^{\mathrm{P}}_{opp}.\, \varphi \in C(\gamma) \wedge \varphi$ *variant* conflicts with α **do**
        RefuteRule(φ, O)
      end
    end
  end
end

Procedure CheckRule is invoked when we are trying to prove a certain rule α, with modality X, and we found an applicable and proved rule β supporting α. To verify that the opposite rules are defeated, we need to distinguish between the two conflict variants. If $variant = simply$, then $R[\alpha]^X_{infd}$ will contains all those γ rules that simply conflict with and are defeated by β. If $variant = rationally$, we need to consider more rules: not just the γs defeated by β, but also those γs *not* stronger than β such that their conclusion is defeater by α itself. This perfectly mirrors Condition (2.2.2.2) of Definitions 21, 22, and 23.

Lastly, the way of how matrices $\alpha[][]$ are updated at Lines 7 and 18 of Procedure ProveRule (resp. Lines 10 and 24 of Procedure RefuteRule) are because here we have to satisfy Condition 4.b of Definition 14 (and *not* Condition 4.a of Definition 15).

Procedure ProveRule(α, X)

Input: A standard rule α, a modality $X \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$

1 $\partial^+_{X,meta} \leftarrow \partial^+_{X,meta} \cup \{\alpha\}$;
2 $MHB \leftarrow MHB \setminus \{X\alpha\}$;
3 **switch** *modality* X **do**
4 **case** $X = \mathrm{C}$ **do**
5 **for** $\Box p \in MHB$ **do** $R^{\Box}[p] \leftarrow \{A(\beta) \setminus \{\alpha\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[p]\}$;
6 **for** $\Box\zeta \in MHB$ **do** $R^{\Box}[\zeta] \leftarrow \{A(\beta) \setminus \{\alpha\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[\zeta]\}$;
7 **for** $\beta \in R^{\mathrm{O}}[\alpha, i]$ **do** $\beta[2][i] \leftarrow -$;
8 **end**
9 **case** $X = \mathrm{O}$ **do**
10 **for** $\Box p \in MHB$ **do**
11 $R^{\Box}[p] \leftarrow \{A(\beta) \setminus \{\mathrm{O}\alpha, \neg\mathrm{O}{\sim}\alpha, \neg\mathrm{P}{\sim}\alpha\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[p]\} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg\mathrm{O}\alpha\} \in A(\beta)\}$;
12 **end**
13 **for** $\Box\zeta \in MHB$ **do**
14 $R^{\Box}[\zeta] \leftarrow \{A(\beta) \setminus \{\mathrm{O}\alpha, \neg\mathrm{O}{\sim}\alpha, \neg\mathrm{P}{\sim}\alpha\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[\zeta]\} \setminus \{\beta \in R^{\Box}[\zeta] | \, \{\neg\mathrm{O}\alpha\} \in A(\beta)\}$;
15 $R[\zeta]^{\Box}_{opp} \leftarrow R[\zeta]^{\Box}_{opp} \setminus \{\beta \in R^{\Box}[\zeta] | \, \{\neg\mathrm{O}\alpha\} \in A(\beta)\}$;
16 $R[\zeta]^{\Box}_{supp} \leftarrow R[\zeta]^{\Box}_{supp} \setminus \{\beta \in R^{\Box}[\zeta] | \, \{\neg\mathrm{O}\alpha\} \in A(\beta)\}$;
17 **end**
18 **for** $\beta \in R^{\mathrm{O}}[\alpha, i]$ **do** $\beta[1][i] \leftarrow +$;
19 **end**
20 **case** $X = \mathrm{P}$ **do**
21 **for** $\Box p \in MHB$ **do**
22 $R^{\Box}[p] \leftarrow \{A(\beta) \setminus \{\mathrm{P}\alpha, \neg\mathrm{O}{\sim}\alpha\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[p]\} \setminus \{\beta \in R^{\Box}[p] | \, \{\neg\mathrm{P}\alpha\} \in A(\beta)\}$;
23 **end**
24 **for** $\Box\zeta \in MHB$ **do**
25 $R^{\Box}[\zeta] \leftarrow \{A(\beta) \setminus \{\mathrm{P}\alpha, \neg\mathrm{O}{\sim}\alpha\} \hookrightarrow_{\blacksquare} C(\beta) | \, \beta \in R^{\Box}[\zeta]\} \setminus \{\beta \in R^{\Box}[\zeta] | \, \{\neg\mathrm{P}\alpha\} \in A(\beta)\}$;
26 $R[\zeta]^{\Box}_{opp} \leftarrow R[\zeta]^{\Box}_{opp} \setminus \{\beta \in R^{\Box}[\zeta] | \, \{\neg\mathrm{P}\alpha\} \in A(\beta)\}$;
27 $R[\zeta]^{\Box}_{supp} \leftarrow R[\zeta]^{\Box}_{supp} \setminus \{\beta \in R^{\Box}[\zeta] | \, \{\neg\mathrm{P}\alpha\} \in A(\beta)\}$;
28 **end**
29 **end**
30 **end**

Procedure RefuteRule(α, X)

Input: A standard rule $\alpha : A(\alpha) \hookrightarrow_{\Box} C(\alpha)$, a modality $X \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$

$\partial^{-}_{X,meta} \leftarrow \partial^{-}_{X,meta} \cup \{\alpha\}$;
$MHB \leftarrow MHB \setminus \{X\alpha\}$;
if $X = \mathrm{C}$ **then**
  for $\Box p \in MHB$ **do** $R^{\Box}[p] \leftarrow R^{\Box}[p] \setminus \{\beta \in R^{\Box}[p] |\, \alpha \in A(\beta)\}$;
  for $\Box\zeta \in MHB$ **do**
    $R^{\Box}[\zeta] \leftarrow R^{\Box}[\zeta] \setminus \{\beta \in R^{\Box}[\zeta] |\, \alpha \in A(\beta)\}$;
    $R[\zeta]^{\Box}_{opp} \leftarrow R[\zeta]^{\Box}_{opp} \setminus \{\beta \in R^{\Box}[\zeta] |\, \alpha \in A(\beta)\}$;
    $R[\zeta]^{\Box}_{supp} \leftarrow R[\zeta]^{\Box}_{supp} \setminus \{\beta \in R^{\Box}[\zeta] |\, \alpha \in A(\beta)\}$;
  end
  for $\beta \in R^{\mathrm{O}}[\alpha, i]$ **do** $\beta[2][i] \leftarrow +$;
  $> \leftarrow > \setminus \{(\beta, \gamma), (\gamma, \beta) \in > |\, \alpha \subseteq A(\beta)\}$;
else
  for $\Box p \in MHB$ **do**
    $R^{\Box}[p] \leftarrow \{A(\beta) \setminus \{\neg X\alpha\} \hookrightarrow_{\blacksquare} C(\beta) |\, \beta \in R^{\Box}[p]\} \setminus \{\beta \in R^{\Box}[p] |\, X\alpha \in A(\beta)\}$;
  end
  for $\Box\zeta \in MHB$ **do**
    $R^{\Box}[\zeta] \leftarrow \{A(\beta) \setminus \{\neg X\alpha\} \hookrightarrow_{\blacksquare} C(\beta) |\, \beta \in R^{\Box}[\zeta]\} \setminus \{\beta \in R^{\Box}[\zeta] |\, X\alpha \in A(\beta)\}$;
    $R[\zeta]^{\Box}_{opp} \leftarrow R[\zeta]^{\Box}_{opp} \setminus \{\beta \in R^{\Box}[\zeta] |\, X\alpha \in A(\beta)\}$;
    $R[\zeta]^{\Box}_{supp} \leftarrow R[\zeta]^{\Box}_{supp} \setminus \{\beta \in R^{\Box}[\zeta] |\, X\alpha \in A(\beta)\}$;
  end
  $> \leftarrow > \setminus \{(\beta, \gamma), (\gamma, \beta) \in > |\, X\alpha \subseteq A(\beta)\}$;
end
if $X = \mathrm{O}$ **then**
  for $\beta \in R^{\mathrm{O}}[\alpha, i]$ **do** $\beta[1][i] \leftarrow -$;
end

Algorithms Execution

We end this part by proposing a couple of input theories and analysing the runs of the algorithms in the computation of the extension. Each theory tackles specific characteristics of our algorithms and therefore of our logic. We will pay attention only to the relevant and non-trivial details.

Consider $D = (\{f_1, f_2\}, R, \{(\gamma, \theta)\})$ to be a theory such that

$$R = \{\alpha\colon (\gamma\colon \neg f_1 \Rightarrow_{\mathrm{C}} a) \Rightarrow_{\mathrm{C}} b \qquad \beta\colon f_2 \Rightarrow_{\mathrm{C}} (\gamma\colon \neg f_1 \Rightarrow_{\mathrm{C}} a)$$
$$\zeta\colon (\nu\colon f_1 \Rightarrow_{\mathrm{C}} c) \Rightarrow_{\mathrm{C}} (\kappa\colon f_2 \Rightarrow_{\mathrm{P}} \neg a) \qquad \theta\colon f_1, f_2 \Rightarrow_{\mathrm{C}} \neg a$$
$$\mu\colon f_2 \Rightarrow_{\mathrm{O}} a \otimes b \otimes c\},$$

Support sets $R^{\Box}[l]$ are instantiated; what is relevant here are $R^{\mathrm{C}}[a] = \{\gamma\}$, $R^{\mathrm{C}}[\neg a] = \{\theta\}$, $R^{\mathrm{O}}[a] = \{\mu\}$, and $R^{\mathrm{P}}[\neg a] = \{\kappa\}$. The only obligation matrix created is $\mu[][]$ (2×3), as μ is the only obligation rule.

Procedure ProveLiteral is invoked on f_1 and f_2, which results in: (i) both literals are added to $+\partial_{\mathrm{C}}$, (ii) emptying $A(\beta)$, $A(\nu)$, $A(\kappa)$, $A(\theta)$, and (iii) $A(\mu)$.

Procedure RefuteLiteral is symmetrically invoked on $\neg f_1$ and $\neg f_2$, which results in: (i) removing γ from $R^{\mathrm{C}}[a]$ as this makes γ to be discarded (note that now there are no more supporting rules to conclude a as constitutive), and (ii) removing (γ, θ) from the superiority relation. Important to highlight here that, even if γ is discarded, this does not influence the applicability/discardability of α, which depends only on proving either $+\partial^m_{\mathrm{C}}\gamma$ or $-\partial^m_{\mathrm{C}}\gamma$. Rules α, β, ζ, θ, and μ are all proved as constitutive (thus $\alpha, \beta, \zeta, \theta, \mu \in +\partial^m_{\mathrm{C}}$).

The algorithm finally enters the main **repeat**–**until** cycle; we shall assume that the loop **for** at Lines 19–29 first controls whether we can prove b as constitutive. Indeed there exists a constitutive rule for b that has been proved: α. Procedure CheckLiteral is thus invoked with b, C, and α, as parameters. The first half of the check $\{\, R[\neg b] \setminus R[\neg b]_{infd} = \emptyset \wedge \exists \zeta \in R^{\mathrm{C}}_{\mathrm{d}}[l].\, A(\zeta) = \emptyset \,\}$ is in fact satisfied as there are no rules supporting $\neg b$, but not the second half as applicability/discardability of α has yet to be determined; anyhow, $\neg b$ has no support and Procedure RefuteLiteral is hence invoked, with the only result to eliminate it from *MHB* and add it to $-\partial_{\mathrm{C}}$.

Assume that the loop **for** at Lines 19–29 now checks $\neg a$ for constitutive; θ is applicable and no rules support a (we do not need to wait for γ to be proved/rejected as we already know that, if proved, γ is discarded as previously we computed that $\neg f_1 \in -\partial_{\mathrm{C}}$). Accordingly, Procedure ProveLiteral: (i) computes $\neg a \in +\partial_{\mathrm{C}}$, and (ii) updates $\mu[2][1]$ to $+$. Analogously to the previous case, we cannot yet decide for a as obligation since there is potentially the permissive rule κ for $\neg a$, and we thus need to wait until we compute either $\pm\partial_{\mathrm{C}}\kappa$.

The algorithm passes then to run the loop **for** at Lines 32–42, and let us consider γ; β is applicable and no rule conflicts with it. We hence compute $\gamma \in +\partial^{meta}_{\mathrm{C}}$, which in turn updates $\{\, A(\alpha) \setminus \{\alpha\} = \emptyset \,\}$ (α is now applicable). On the contrary, no rules support ν, thus in cascade: (i) $\nu \in -\partial^{meta}_{\mathrm{C}}$, (ii) ζ is discarded, hence (iii) no rules support κ and so $\kappa \in -\partial^{meta}_{\mathrm{C}}$.

At the next iteration of the **repeat**–**until**, (i) we compute $b \in +\partial_{\mathrm{C}}$ and update $\mu[2][2]$ to $-$, (ii) $a \in +\partial_{\mathrm{O}}$ and we update $\mu[1][1]$ to $+$. This makes μ applicable at index 2 for b, and we compute $b \in +\partial_{\mathrm{O}}$. Lastly, as $\mu[1][2] = +$ and $\mu[2][2] = -$ makes μ discarded at index 3 for c, the algorithm computes $c \in -\partial_{\mathrm{O}}$.

The next example illustrates: (i) how the algorithms work on the chains of meta-rules, and (ii) the computation of the supporting sets *opp* and *supp* for the rational conflict variant, including cases where the conflict is not restricted to the negations of the rules.

Consider $D = (F, R, > = \{(\alpha, \beta), (\alpha, \lambda)\})$ to be a theory such that

$$
\begin{aligned}
R = \{ & \alpha\colon \ldots \Rightarrow_{\mathrm{O}} (\eta\colon a \Rightarrow_{\mathrm{P}} b) \otimes c \otimes (\kappa\colon c \Rightarrow_{\mathrm{O}} d \otimes e) \\
& \beta\colon \ldots \Rightarrow_{\mathrm{P}} (\theta\colon a \Rightarrow_{\mathrm{O}} \neg b) \qquad \gamma\colon \ldots \Rightarrow_{\mathrm{O}} \neg(\zeta\colon a \Rightarrow_{\mathrm{O}} \neg b) \\
& \lambda\colon \ldots \Rightarrow_{\mathrm{O}} (\mu\colon c \Rightarrow_{\mathrm{O}} d) \qquad \psi\colon \ldots \Rightarrow_{\mathrm{C}} \neg c\},
\end{aligned}
$$

where for the sake of simplicity we assume that all the rules' antecedents have been emptied, and would thus have satisfied Conditions 1–3 of Definition 14. According to Definitions 16 and 17, we notice that:

1. η rationally conflicts with θ, hence α conflicts with β (at index 1);
2. θ simply conflicts with ζ, hence β conflicts with γ;
3. For 1 and 2, γ supports α (and the other way around);
4. κ rationally conflicts with μ, hence λ rationally conflicts with α (at index 3).

Ergo, $R[\alpha]^{\mathrm{O}}_{opp} = \{\beta, \lambda\}$, $R[\alpha]^{\mathrm{O}}_{supp} = \{\gamma\}$, and so on. When Procedure CheckRule is invoked on η, **if** check of Procedure ProveRule at Line 26 is satisfied we compute $\eta, \neg\zeta \in +\partial^{meta}_{\mathrm{O}}$, $\theta \in -\partial^{meta}_{\mathrm{P}}$ as well as $\theta \in -\partial^{meta}_{\mathrm{O}}$.

As later on, the algorithms compute $\neg c \in +\partial_{\mathrm{C}}$ as well as $c \in +\partial_{\mathrm{O}}$ (and thus $\alpha[j][2]$, $j = 1, 2$, are updated to $+$), then α becomes applicable at index 3: given that $\alpha{>}\lambda$, we compute $\kappa \in +\partial^{meta}_{\mathrm{O}}$ and $\mu \in -\partial^{meta}_{\mathrm{O}}$.

Note that, if we had considered the simple conflict variant, α would have not conflicted with β and since the superiority relation does not solve the conflict between β and γ, we would have computed $\neg\zeta \in -\partial^{meta}_{\mathrm{O}}$ instead of $\neg\zeta \in +\partial^{meta}_{\mathrm{O}}$.

6.1 Computational Properties

We discuss the computational properties of the algorithms presented. In order to discuss termination and computational complexity, we start by defining the *size* of a meta-theory D as $\Sigma(D)$ to be the number of the occurrences of literals plus the number of occurrences of rules plus 1 for every tuple in the superiority relation. Consequently, theory $D = (F = \{a, b, c\}, R = \{(\alpha\colon a \Rightarrow_{\mathrm{O}} d), (\beta\colon b \Rightarrow \sim d), \big(\gamma\colon c \Rightarrow (\zeta\colon a \Rightarrow d)\big)\}, > = \{(\zeta, \beta)\})$ has size $3 + 11 + 1 = 15$.

Note that, by implementing hash tables with pointers to rules where a given literal occurs, each rule can be accessed in constant time. We also implement hash tables for the tuples of the superiority relation where a given rule appears as either of the two element, and even those can be accessed in constant time.

Lemma 1. *Procedures ProveLiteral, RefuteLiteral, ProveRule, and RefuteRule terminate and their complexity is* $O(\Sigma^2)$.

Proof. Termination of such procedures is straightforward, as (i) the size of the input theory is finite, (ii) we modify finite sets only, and (iii) all the **for** cycles loop on a finite number of elements, as the Modal Herbrand Base is finite.

Given that all set assignments/modifications are linear in the size of the theory, and that all the **for** cycles are iterated $|MHB| \in O(\Sigma)$ times, this proves our claim setting their complexity to $O(\Sigma^2)$.

Lemma 2. *Procedure CheckLiteral terminates and its complexity is* $O(\Sigma^3)$.

Proof. Termination is straightforward (see motivations above) and by the fact that Lemma 1 proves that its inner Procedures ProveLiteral and RefuteLiteral terminate.

Again, all set assignments/modifications are linear in the size of the theory, and so are all the **if** checks that which invoke Procedures ProveLiteral and RefuteLiteral. Accordingly, $O(\Sigma) + O(\Sigma) * O(\Sigma^2) = O(\Sigma^3)$.

Lemma 3. *Procedure Conflicts terminates and its complexity is:* $O(\Sigma^3)$ *if* $variant = simply$*, or* $O(\Sigma^4)$ *if* $variant = rationally$*.*

Proof. Termination is guaranteed by the same considerations above, and from what follows, depending on the variant.

If $variant = simply$, set assignments operate on finite sets and: (i) R^X_{opp} are linear, (ii) R^X_{supp} are in $O(\Sigma^2)$. This sets the overall complexity to $O(\Sigma^3)$.

If $variant = rationally$, we report hereafter how a set R^X_{opp} is.

$$\begin{aligned} R[\alpha]^X_{opp} \leftarrow R[\alpha]^X_{opp} \cup \{\gamma \in R[\varphi] |\, & C(\alpha) = c_1 \otimes \cdots \otimes c_m \\ & \wedge\, C(\varphi) = d_1 \otimes \cdots \otimes d_n \wedge \big(A(\alpha) = A(\varphi)\big) \wedge \\ & \big(\exists i \leq m, n. \forall k < i. (c_k = d_k \wedge c_i = {\sim} d_i) \\ & \vee (m < n \wedge \forall j \leq m. c_j = d_j)\big)\}. \end{aligned}$$

Such a set is finite, and is computed in $O(\Sigma^2)$ (which implies that R^X_{supp} is in $O(\Sigma^3)$). As the main **for** cycle at Lines 1–35 is in $O(\Sigma)$, this proves our claim.

As before, Procedure Conflicts uses hash tables to store such information, and so after its iteration to verify whether a certain rule *variant* conflicts with another requires constant time, whereas to identify all the rules that *variant* conflict with a certain rule requires linear time.

Lemma 4. *Procedure CheckRule terminates and its complexity is* $O(\Sigma^4)$*.*

Proof. Termination is straightforward (see motivations above) and given by the fact that Lemma 1 proves that its inner Procedures ProveRule and RefuteRule terminate.

Again, all set assignments/modifications are linear in the size of the theory, even the ones requiring the verification of *variant conflicts with* based on the considerations in (and after) Lemma 3.

Given that all the **if** checks and the **for** cycles are linear as well, this proves that the overall complexity is $O(\Sigma) + (\Sigma) * (O(\Sigma^2) + O(\Sigma) * O(\Sigma^2)) = O(\Sigma^4)$.

Theorem 3. *Algorithm 1* Deontic Meta Extension *terminates and its complexity is* $O(\Sigma^5)$*.*

Proof. Termination of Algorithm 1 Deontic Meta Extension is given by Lemmas 1–4, and bound to the termination of the **repeat-until** cycle at Lines 17–44, as all other cycles loop over finite sets of elements of the order of $O(\Sigma)$. Given that (i) the Modal Herbrand Base *MHB* is finite and (ii) since, every time a literal or a rule is proved/refuted, they are removed from *MHB*, thus the algorithm eventually empties such sets, and, at the next iteration, no modification to the extension can be made. This proves the termination of Algorithm 1.

Regarding its complexity, let us notice that: (i) all set modifications are in linear time, and (ii) the aforementioned **repeat-until** cycle is iterated at most $O(\Sigma)$ times, and so are the two **for** loops at lines 19–29 and 32–42. This would

suggest that the **repeat-until** cycle would contribute to the overall complexity for a factor of $O(\Sigma^2)$. A more discerning analysis shows that the complexity is actually $O(\Sigma)$, as the complexity of each **for** loop cannot be considered separately from the complexity of the external loop (they are strictly dependent on one another). Indeed, the overall number of operations made by the sum of all loop iterations cannot outrun the number of occurrences of the literals or rules ($O(\Sigma) + O(\Sigma)$), because the operations in the inner cycles directly decrease, iteration after iteration, the number of the remaining repetitions of the outmost loop, and the other way around.

Based on the results of Lemmas 1–4, and the fact that the complexity of the **repeat-until** cycle, which is $O(\Sigma^5) = O(\Sigma)*(O(\Sigma^2)+O(\Sigma^3)+O(\Sigma^2)+O(\Sigma^4))$, dominates all the operations in the first part of the algorithm, we have that the overall complexity is $O(\Sigma^5)$.

The final result concerns the soundness and correctness of the algorithm, in the sense that the extensions computed by the algorithm corresponds to the set of provable/refutable literals/rules.

Theorem 4. *Algorithm 1* Deontic Meta Extension *is sound and complete, that is, for* $\Box \in \{\mathrm{C}, \mathrm{O}, \mathrm{P}\}$*:*

1. $D \vdash_L +\partial_\Box p$ *iff* $p \in +\partial_\Box p$ *of* $E_L(D)$, $p \in$ Lit
2. $D \vdash_L +\partial^m_\Box \alpha$ *iff* $p \in +\partial^m_\Box \alpha$ *of* $E_L(D)$, $\alpha \in$ Lab
3. $D \vdash_L -\partial_\Box p$ *iff* $p \in -\partial_\Box p$ *of* $E_L(D)$, $p \in$ Lit
4. $D \vdash_L -\partial^m_\Box \alpha$ *iff* $p \in -\partial^m_\Box \alpha$ *of* $E_L(D)$, $\alpha \in$ Lab.

Proof. (Sketch) The aim of Algorithm 1 Deontic Meta Extension is to compute a defeasible meta-extension of the input theory through successive transformations on the set of facts, rules, and the superiority relation. Such transformations allow us: (i) to obtain a simpler theory, (ii) while retaining the same extension. By simpler theory we mean a theory with less symbol in it. Note that if $D \vdash_L +\partial_\Box l$ then $D \vdash_L -\partial_L \sim l$, and that if $D \vdash_L +\partial^m_\Box \alpha$ then $D \vdash_L -\partial^m_\Box \gamma$, with α conflicting with γ.

Suppose that the algorithm computes $+\partial_\Box l$ or $+\partial^m_\Box \alpha$ (meaning that $l \in +\partial_\Box$, or $\alpha \in +\partial^m_\Box$). Accordingly, we remove $l/\Box l$ or $\alpha/\Box\alpha$ from every antecedent where it appears in as, by Definition 14, the applicability of such rules will not depend any longer on $l/\Box l$ or $\alpha\Box\alpha$, but only on the remaining elements in their antecedents. Moreover, we can eliminate from the rule sets all those rules with $\sim l/\Box\sim l$ or $\gamma/\Box\gamma$ in their antecedent (with α conflicting with γ), as such rules are discarded by Definition 15 (and then adjust the superiority relation accordingly). Finally, proving $+\partial^m_{\mathrm{C}} \alpha$ makes α to become active in supporting its conclusion, and rebutting the opposite.

The proof follows the schemas of the ones in [35,37], and consists in proving that the meta-extension of the original theory D and the meta-extension of the simpler theory D' are the same.

Formally, suppose that $D \vdash_L +\partial_\Box l$ (symmetrically $D \vdash_L +\partial^m_\Box \alpha$) at $P(n)$. R' of D' is obtained by the following transformation. Given a standard rule

$\alpha\colon A(\alpha) \hookrightarrow_{\blacksquare} C(\alpha)$ and a literal l, $\alpha \ominus l$ is the rule

$$\alpha\colon A(\alpha) \setminus \{l\} \hookrightarrow_{\blacksquare} C(\alpha).$$

Given a meta-rule rule $\alpha\colon A(\alpha) \hookrightarrow_{\blacksquare} C(\alpha)$ and a literal l, $\alpha \ominus l$ is the rule

$$\alpha\colon A(\alpha)[\beta/\beta \ominus l] \setminus \{l\} \hookrightarrow_{\square} C(\alpha),$$

for all rules $\beta \in A(\alpha)$, where $A(\alpha)[\beta/\beta \ominus l]$ denotes the substitution of β in α with $\beta \ominus l$.

Furthermore, if $\alpha \ominus \beta$ is the rule

$$\alpha\colon A(\alpha) \setminus \{\beta\} \hookrightarrow_{\blacksquare} C(\alpha),$$

then[16]

$$R' = \{\alpha \ominus \Box l |\, \alpha \in R\} \setminus \{\alpha \in R |\, {\sim}\Box l \in A(\alpha)\}$$

for $D \vdash_L +\partial_{\square} l$, while for $D \vdash_L +\partial^m_{\square}\alpha$ we use

$$R' = \{\beta \ominus \Box\alpha |\, \beta \in R\} \setminus \{\beta \in R |\, {\sim}\Box\alpha \in A(\beta)\}$$

Finally, if $>'$ of D' is obtained from $>$ of D as follows

$$>' = > \setminus \{(\beta, \zeta), (\zeta, \beta) \mid {\sim}l \in A(\zeta) \text{ or } {\sim}\gamma \in A(\zeta)\},$$

then by induction on the length of a proof, we can show that

- $D \vdash_L \pm\partial_{\square} p$ iff $D' \vdash_L \pm\partial_{\square} p$, and
- $D \vdash_L \pm\partial^m_{\square}\chi$ iff $D' \vdash_L \pm\partial^m_{\square}\chi$.

The key step to proof the equivalences above is that: when we prove l (resp., α), then we can transform a derivation P in D into a derivation P' in D'. We do so by removing from P, first, all steps where l occurs, and then all subsequent steps justified by steps where l occurred. It is immediate to see that if a rule β is applicable in D at a particular step in P, then the version of β in D' (if any) is applicable in D' at some step in P'.

The next proprieties to conclude the proof of the theorem are the following: if R contains a rule α such that $A(\alpha) = \emptyset$, then

1. If $\{\beta \in R |\, \beta \text{ conflicts with } \alpha \wedge \beta > \alpha\} = \emptyset$, then $D \vdash_L -\partial_{\square}{\sim}C(\alpha)$.
2. If $\{\beta \in R |\, \beta \text{ conflicts with } \alpha \wedge \beta > \alpha\} \setminus \{\beta \in R |\, \beta \text{ conflicts with } \alpha \wedge \exists\gamma, \gamma \text{ conflicts with } \beta \wedge A(\gamma) = \emptyset \wedge \gamma > \beta\} = \emptyset$,[17] then $D \vdash_L -\partial_{\square}{\sim}C(\alpha)$.

First of all, α is applicable. Then for 1., it is a witness of an applicable and undefeated rule for ${\sim}C(\alpha)$. It is mundane to verify that it satisfies the $\exists\gamma\ldots$ clause of the proof conditions for the various $-\partial$. Similarly for 2., if all rules conflicting with α have been defeated (the construction, denoted by $R[]_{infd}$ in the algorithms, ensures, that for every rule opposing α is defeated by an applicable rule), satisfying the $\exists\zeta$ clauses of the proof conditions for the various $+\partial$. Condition 1. is used by Procedures RefuteLiteral and RefuteRule, and Condition 2. by the Procedures ProveLiteral and ProveRule to populate the extension of a theory.

[16] If $\square = \mathrm{C}$, in the transformation $\Box l$ is just l, following the conditions of Definition 14 to make a rule applicable, and ${\sim}\Box l$ is ${\sim}l$ according to Definition 15 to discard a rule.

[17] For $L = S$ we require the additional condition that γ is either α or ${\sim}\beta$, and for meta-rules and $L = R$, we have to consider not only the meta-rules stronger than β, but the standard rules stronger than $C(\beta)$.

7 Related Work

The philosophical issue of admitting meta-rules in the logical language was critically discussed for a long time in conditional logics [62]. At the end of the '60s, such logics have been studied to give formal account to linguistic structures such as indicative conditionals and subjunctive conditionals to express counterfactuals. Later on, conditional logics have been also used to deal with non-monotonic reasoning [19,24]. While some, like [1], rejected nested rules when a probabilistic treatment of conditionals is considered, others admitted this possibility even though it was argued that the intuitive meaning of such formulas is not clear [48,62,19]; meaningful use of nested conditionals were advanced in [12].

In mathematical logic and theoretical computer science meta-logic and meta-reasoning have been a discussed theme across the years. An historical perspective towards the argument shows that the topic has been dealt with, in different perspectives, since 1989, when Paulson [69] presented the issues related to the construction of a generic theorem prover and showed that reasoning with the notion of *relevance* as a means to choose the most useful subset of axioms can boost a significant speed-up of the process itself.

Thenceforth, the discussion stood open for a few years. Numerous scholars borrowed the notion of meta-reasoning from the original presentation of the issue proposed by Tarski [78] where the notion of *metamathematics* is devised as applied to logical frameworks. This was employed in the foundational work by Russell and Wefald [72]. The level of discussion provided in Russell and Wefald's study was yet too far from concrete cases, but it has the great merit of providing a reference for the basic idea needed for devising the *motivation* for introducing meta-reasoning systems, defined as *bounded rationality*. More concretely it became possible to employ these notions in many-valued logics [60], in image processing, a field that is not relevant to this investigation, and in knowledge-based systems by Rowel et al. [71].

Almost immediately a significant attention has been posed on the definition of meta-logic methods in logic programming. This investigation line started by the pioneering study by Costantini and Lanzarone [16], but was inspired by the early investigation by Lloyd et al. [51] and further investigated by Brogi et al. [14,13], Lifschitz et al. [49], and many others.

Subsequently, a very important novelty in meta-logical systems was introduced by Grundy in 1996 [43]: the notion of hierarchical reasoning. The idea behind hierarchical reasoning is that the setting of decision making does not depend on one single level, but on more than one. There have been numerous applications that have been devised, and in many of these hierarchical reasoning has shown to be decisive. Papadias et al. [68] provided specific algorithms for spatial reasoning, and Seidel et al. [74] applied it to probabilistic Constraint Satisfaction Problems.

Differently than in hierarchical reasoning, where different layers of logical frameworks are employed to represent hierarchy of decisions, metareasoning can be based on higher order logics, where reasoning becomes an element of the logic itself. This approach is the inspiration of methods for the construction

of systems with nested rules, that we introduced as a base for this research in Section 1. Original investigation on higher-order logic and metareasoning should be attributed to McDowell et al. [57] and independently to Momigliano et al. [59].

Meta-reasoning has also been the motivation for investigations on a more theoretical level. On the one hand, the notion of continual computation has been deeply investigated as a base for machine reasoning by Horvitz [45]. His revolutionary notion has shown to be the base of many unified models of reasoning for humans and machines, as discussed widely by Gershman et al. in the impactful study on the notion of computational rationality [28]. Another aspect of meta-logic that has been dealt with is the *temporal* one, that has been studied for the rewriting methods by Clavel and Meseguer [15], and by Baldan et al. [5].

To come to meta-reasoning and meta-logic in non monotonic systems, this work is an extensive revision, and substantial extension to the research by Olivieri et al. [64]. In that work, authors have provided a framework for reasoning with meta-rules, as a generalization of the approach employed by Governatori and Rotolo [31] and by Cristani *et al.* [17]. These studies have been mainly focused upon the needs for specific meta-rules for the revision of preferences as discussed by Governatori *et al.* [30], and to accommodate, in form of meta-reasoning, the issues related to effects of rules over time. In the specific aspects related to Business Process Compliance some studies have been developed by Governatori et al. [37,35,35,63] where some of the issues that we solved here emerged at first.

The specific issue in developing methods for reasoning with rules as object of an argument has been the focus of the study by Modgil et al. [58]. In this research the focus has mainly been on meta-arguments, that have been investigated as base for a general theory of argumentation structures. Finocchiaro has explored critically this issues [25] and interpreted the notion of meta-argument from a general point of view. Similarly Dhovhannisyan and Djijian [46] recently developed a general theory of meta-argumentation. On the other hand, Boella et al. have defined when a meta-argument is acceptable [11].

8 Conclusions and Further Developments

The focus of this paper is the efficient computation of rules from meta-rules. In general, the topic of how to use (meta-)rules to generate other rules has received little attention, as discussed in Section 7. In this section we refer some works of the late literature that constitutes the terrain on which we build this study.

As previously claimed meta-rules have been considered as a base for norm modification [41,17], and more generally, in [75], that is dedicated to a logic for deriving rules from meta-rules; however, none of these works investigated the computationally complexity, nor addressed the issue of defining algorithms.

The large majority of the studies that have made use of meta-rules have focused upon the usage of these as a means to determine the *scope* of rule application, or the *result* of the application of the rules. In particular, we can

identify two research lines: Logic Programming, and Meta-logic. Logic programming studies investigated the issue of enhancing the expressiveness by allowing nested expressions [50,49]. Nevertheless, these approaches are based on the so-called Lloyd-Toper transformation, that transforms nested expressions into (equivalent) logical expressions. Similarly, in [42] disjunctive DATALOG is enriched with nested rules; however, such nested rules may be eliminated by using (stratified) negation, but these are kept because they allow for a more natural correspondence with the natural language description of the underlying problem.

We have seen in Section 1 that this approach suffers from several problems, and it is not appropriate for many uses of meta-rules, in particular when the aim is to represent meta-rules as means to derive rules. Some works, like [26], extended Logic Programming with negation with nested hypothetical implications, plus constraints or negation as failure. Specifically, they consider rules with conditional goals in the body, but not implications in the head.

The notion of meta-rules and close concepts, including meta-logic [8] and meta-reasoning [22], have been employed widely in Logic Programming [4], but also outside it, specifically in *Context Theory* [29]. In general, we can look at these studies as methodologically coherent with the notion of *hierarchical* reasoning, where it is devised a method to choose which reasoning process is more appropriate for the specific scenario in which the process is employed. A specific line of research (strictly connected with the studies in the semantics of Logic Programming) is the Answer Set Programming (ASP) and preferences [23]. Further on, many studies on ASP where meta-rules took place: unfortunately, such investigations did not focus upon *nested rules*.

Generation of rules from other rules is one of the goals of the research in Input/Output logic (IOL) [54]. The idea of Input-Output Logic is to define a set of operations on input/output rules (where an input/output rule is a pair (x, y), where x and y are formulas in a logical language) to derive new input/output pairs. Differently to what we do (1): IOL does not consider nested rules, and (2) the derivation mechanism depends on the properties of the operations on which the variant of IOL is defined, and not on the rules on which the logic operates.

Closer to the issues of Deontic Logic are the researches in *argumentation*. The basic concept derived by the combination of meta-logical structures and argumentation is the meta-level argumentation [58]. Applied meta-level has been investigated in the view of developing a framework where, for instance, admissibility of arguments, and other issues in this field, are dealt with [21].

Although some previous work has been done, the problem of nested rules in non-monotonic frameworks from a computational complexity viewpoint deserves a deeper study, and this paper fills this gap. In [64], we have been dealing with Defeasible Logic without modal operators and temporal expressions (most of the work on meta-rules considers combinations of such features). In the present work, we have been discussing the modalised formulae and their usage in meta-rules.

Further direct issues concern the introduction of temporal operators: as discussed in [56], applicability issues are strongly concerned with time, and, for what concerns the specific of private international law, also space (wrt countries

and territories). We plan to extend and combine the algorithms presented here with the algorithms for temporal and spatial logical framework with defeasible deontic meta-logic, and we expect that the computational results to carry over.

9 Review1

General assessment: The 'Deontic Meta Rules' is a relatively extensive paper. In case of a 50 pages long paper the authors should allocate dedicated attention to an easy-to-follow structure.

Some traces of such an effort can be found however there is clearly room for improvement in this area: it is often visible that the parts are cut and pasted from different papers or versions lacking a comprehensive flow. Together with the issues indicated below in detail, I believe the paper needs some revisions in order to get publishable.

Detailed comments (technical, substantial, phrasing, and typos miscellaneously)

The abstract has some misleading suggestions. The topic indeed has some connection to the issue of power, however, the authors do not discuss it at all, and while they suggest that their theory will be "to tackle" it, they provide no insight at all how – which is not strange as they do not consider agents or the theory of normative positions at all, without which it would be difficult to say anything substantial about power. This is of course not a problem, if this is not the goal of the authors and their present endeavor (and the paper is quite extensive anyway), but then this "promise" has no place in the abstract.

On p2, last section, from the sentence starting with "Another situation" a new paragraph should start.

DONE

On p5, the last paragraph the authors say that the weak permission "corresponds to saying that something is allowed by a code only when it is not prohibited by that code". I believe there is no need for 'only', it suggests that if there is strong permission, weak permission doesn't follow, while it does. In the next sentence, before the last two formulae: I suppose it's supposed to be "does not support the derivation of either" and not "...both" as it is now.

DONE

On p6, the authors claim that (1) is a desirable basic property. It is, if box=O, but (1) is not desirable if box=P. Just imagine a situation in which the EU prescribes that a member state has to remain neutral regarding a situation (refrain from regulating or declaring both b and not b are permitted). On the other hand, the EU might itself remain neutral regarding an obliging or prohibiting rule. If so, (4) and (5) on p6 are not in conflict. The authors keep on calling the approach in which they are in conflict 'rational' but I don't think that we should consider such declaration of being neutral count as irrational. Maybe another world would express more precisely what the difference is between these two approaches.

RESPONSE: Many thanks for the comment. We have rewritten that part and added a broader discussion.

On p7, in the last paragraph of Section 2, the last but one sentence should be rewritten as a connective (maybe 'if) is missing. Also, here, I believe 'introduce' should be used instead of 'enforce'.

DONE

In Definition 1, I don't think we need the first comma.

DONE

On p8 and p9 (maybe later too) the authors use the 'prescriptive behavior' expression. I suggest using 'utterance' instead.

On p9, it doesn't seem to be fit to declare any plain literal an ox-expression as later plain literals and ox-expressions are handled as different cases like on p10 in Definition 2 5. Right after this, the authors say that in this paper they will only consider ox-expression with only one elementCESK: Davvero?. But then why do we consider ox-expressions with many elements on p10 in the paragraph after Definition 2? In the next paragraph the authors use the expression 'head'. While with some formalisms it is somewhat common to use it, it hasn't been introduced in this paper.

RESPONSE: We do not declare a plain literal as an ox-expression. We give a recursive definition of ox-expressions, where the base clause where we say that an ox-expression (of length one) consists of a plain literal. ox-expressions can only occur in the conclusion (head) of a prescriptive rule; thus, even in the case where we have an ox-expression in the conclusion of a prescriptive rule there is no risk of confusion. Finally, the conclusion of of a prescriptive rule is a single ox-expression. However, there are no constraints on its length (number of elements in it).

In footnote 8, the authors explain the last part of condition 4 of Definition 5. I agree with that the first option implies the second one. Exactly that's the reason for having the second option in the formula and not the first one! CESK: I am missing his point here...

RESPONSE: The first option is more general (since it implies the second); the second is more specific, being implied by the first. As discussed in CITE they can have different applications (and this paper is not the venue to discuss them). Also, we believe the first option is, arguably, more natural. Suppose you have that A is obligatory. We say that we have the violation when NOT A hold. The second option corresponds to say that the obligation of A have been violated when we do not know if A hold.

In the last sentence above Definition 3, instead of the comma it should be inserted that 'means that' to make the sentence readable.

NEED TO REVISE THIS.

P11, first paragraph, 4th line: 'have assert'?

P11, first paragraph, 10th line: it – > if

P12, last sentence above Definition 7: "... that _they/those_ are obtained""

Footnote 8: "we have two option_s_" There is some randomness in the references to the about how to use 'meta': I find the meta-DDL on p19 if the name of

the logic is not Meta Defeasible Deontic Logic but Defeasible Deontic Meta-Logic as it is indicated in the title of Section 4.

The first paragraph is Section 4 on p16 is letter by letter on p7 already.

P16, in the "On the one hand" sentence: "simply extend" (no s)

P19 in 4.1, in the 8th line it should be 'no' and not 'not'

P27 first paragraph, 3rd sentence: one of the_m_ is applicable The related work section opens the scope unexpectedly: while the issue of meta-reasoning is relevant of course, the emphasis at the beginning of the article is put on a much narrower focus. I believe some balance is needed here.

The Related work and the Conclusions and further developments(/work) sections are not separated properly (and the conclusions is very minimal), so I suggest merging the two and put some flow in them.

References

1. Adams, E.W.: The logic of conditionals : an application of probability to deductive logic / Ernest W. Adams. D. Reidel Pub. Co Dordrecht, Holland ; Boston (1975)
2. Alchourrón, C.E., Bulygin, E.: Normative Systems. Springer Verlag, Berlin (1971)
3. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: Representation results for defeasible logic. ACM Trans. Comput. Log. **2**(2), 255–287 (2001). https://doi.org/10.1145/371316.371517
4. Azab, K., Habel, A.: High-level programs and program conditions. LNCS **5214 LNCS**, 211–225 (2008). https://doi.org/10.1007/978-3-540-87405-8_15
5. Baldan, P., Mancarella, P., Raffaetà, A., Turini, F.: Mutaclp: A language for temporal reasoning with multiple theories. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **2408**(PART2), 1–40 (2002)
6. Barcan Marcus, R.: Iterated deontic modalities. Mind **75**(300), 580–582 (1966)
7. Barcan Marcus, R.: The iteration of deontic modalities. Logique et Analyse **9**, 197–209 (1966)
8. Basin, D., Clavel, M., Meseguer, J.: Reflective metalogical frameworks. ACM Transactions on Computational Logic **5**(3), 528–576 (2004). https://doi.org/10.1145/1013560.1013566
9. Billington, D.: Defeasible Logic is Stable. Journal of Logic and Computation **3**(4), 379–400 (1993). https://doi.org/10.1093/logcom/3.4.379
10. Boella, G., van der Torre, L.: Permissions and obligations in hierarchical normative systems. In: ICAIL '03. pp. 109–118. ACM (2003)
11. Boella, G., Van Der Torre, L., Villata, S.: On the acceptability of meta-arguments. In: Proceedings - 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2009. vol. 2, pp. 259–262 (2009). https://doi.org/10.1109/WI-IAT.2009.159
12. Boutilier, C.: Conditional Logics for Default Reasoning and Belief Revision. Ph.D. thesis, University of British Columbia, Department of Computer Science, 6356 Agricultural Road Vancouver, B.C. Canada (1992)
13. Brogi, A., Contiero, S.: A program specialiser for meta-level compositions of logic programs. New Generation Computing **16**(2), 123–161 (1998). https://doi.org/10.1007/BF03037314

14. Brogi, A., Contiero, S., Turini, F.: Programming by combining general logic programs. Journal of Logic and Computation **9**(1), 7–24 (1999). https://doi.org/10.1093/logcom/9.1.7
15. Clavel, M., Meseguer, J.: Reflection in conditional rewriting logic. Theoretical Computer Science **285**(2), 245–288 (2002). https://doi.org/10.1016/S0304-3975(01)00360-7
16. Costantini, S., Lanzarone, G.: A metalogic programming approach: Language, semantics and applications. Journal of Experimental and Theoretical Artificial Intelligence **6**(3), 239–287 (1994). https://doi.org/10.1080/09528139408953789
17. Cristani, M., Olivieri, F., Rotolo, A.: Changes to temporary norms. In: Proceedings of the International Conference on Artificial Intelligence and Law. pp. 39–48 (2017). https://doi.org/10.1145/3086512.3086517
18. Dastani, M., Governatori, G., Rotolo, A., Song, I., van der Torre, L.: Contextual agent deliberation in defeasible logic. In: PRIMA 2007. LNAI, vol. 5044, pp. 98–109. Springer, Heidelberg (2007)
19. Delgrande, J.P.: A first-order conditional logic for prototypical properties. Artificial Intelligence **33**, 105–130 (1987)
20. Dung, P.M., Sartor, G.: The modular logic of private international law. Artif. Intell. Law **19**(2-3), 233–261 (2011). https://doi.org/10.1007/s10506-011-9112-5, http://dx.doi.org/10.1007/s10506-011-9112-5
21. Dupin De Saint-Cyr, F., Bisquert, P., Cayrol, C., Lagasquie-Schiex, M.C.: Argumentation update in yalla (yet another logic language for argumentation). International Journal of Approximate Reasoning **75**, 57–92 (2016). https://doi.org/10.1016/j.ijar.2016.04.003
22. Dyoub, A., Costantini, S., De Gasperis, G.: Answer set programming and agents. Knowledge Engineering Review **33**(1) (2018). https://doi.org/10.1017/S0269888918000164
23. Eiter, T., Faber, W., Leone, N., Pfeifer, G.: Computing preferred answer sets by meta-interpretation in answer set programming. Theory and Practice of Logic Programming **3**(4-5), 463–498 (2003). https://doi.org/10.1017/S1471068403001753
24. Farinas del Cerro, L., Herzig, A., Lang, J.: From ordering-based nonmonotonic reasoning to conditional logics. Artificial Intelligence **66**, 375–393 (1994)
25. Finocchiaro, M.: Arguments, meta-arguments, and metadialogues: A reconstruction of krabbe, govier, and woods. Argumentation **21**(3), 253–268 (2007)
26. Gabbay, D.M., Giordano, L., Martelli, A., Olivetti, N.: A language for handling hypothetical updates and inconsistency. Log. J. IGPL **4**(3), 385–416 (1996). https://doi.org/10.1093/jigpal/4.3.385, https://doi.org/10.1093/jigpal/4.3.385
27. Gelati, J., Governatori, G., Rotolo, A., Sartor, G.: Normative autonomy and normative co-ordination: Declarative power, representation, and mandate. Artificial Intelligence and Law **12**(1-2), 53–81 (2004)
28. Gershman, S., Horvitz, E., Tenenbaum, J.: Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. Science **349**(6245), 273–278 (2015). https://doi.org/10.1126/science.aac6076
29. Ghidini, C., Giunchiglia, F.: Local models semantics, or contextual reasoning = locality + compatibility. Artificial Intelligence **127**(2), 221–259 (2001). https://doi.org/10.1016/S0004-3702(01)00064-9
30. Governatori, G., Olivieri, F., Cristani, M., Scannapieco, S.: Revision of defeasible preferences. International Journal of Approximate Reasoning **104**, 205–230 (2019)

31. Governatori, G., Rotolo, A.: Changing legal systems: Legal abrogations and annulments in defeasible logic. Logic Journal of the IGPL **18**(1), 157–194 (2009). https://doi.org/10.1093/jigpal/jzp075
32. Governatori, G.: Burden of compliance and burden of violations. In: Rotolo, A. (ed.) 28th Annual Conference on Legal Knowledge and Information Systems. pp. 31–40. Frontieres in Artificial Intelligence and Applications, IOS Press, Amsterdam (2015)
33. 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)
34. Governatori, G., Olivieri, F.: Unravel legal references in defeasible deontic logic. In: Maranhão, J., Wyner, A.Z. (eds.) ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021. pp. 69–78. ACM (2021), https://doi.org/10.1145/3462757.3466080
35. Governatori, G., Olivieri, F., Rotolo, A., Scannapieco, S.: Computing strong and weak permissions in defeasible logic. J. Philos. Log. **42**(6), 799–829 (2013). https://doi.org/10.1007/s10992-013-9295-1, https://doi.org/10.1007/s10992-013-9295-1
36. Governatori, G., Olivieri, F., Scannapieco, S., Cristani, M.: Designing for compliance: Norms and goals. In: RuleML 2011-America. LNCS, vol. 7018, pp. 282–297. Springer (2011). https://doi.org/10.1007/978-3-642-24908-2_29
37. Governatori, G., Olivieri, F., Scannapieco, S., Rotolo, A., Cristani, M.: The rationale behind the concept of goal. Theory Pract. Log. Program. **16**(3), 296–324 (2016). https://doi.org/10.1017/S1471068416000053, https://doi.org/10.1017/S1471068416000053
38. 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). https://doi.org/10.1093/jigpal/jzp006
39. Governatori, G., Rotolo, A.: Logic of violations: A Gentzen system for reasoning with contrary-to-duty obligations. Australasian Journal of Logic **4**, 193–215 (2006), http://ojs.victoria.ac.nz/ajl/article/view/1780
40. Governatori, G., Rotolo, A.: BIO logical agents: Norms, beliefs, intentions in defeasible logic. Auton. Agents Multi Agent Syst. **17**(1), 36–69 (2008). https://doi.org/10.1007/s10458-008-9030-4, https://doi.org/10.1007/s10458-008-9030-4
41. Governatori, G., Rotolo, A.: Changing legal systems: legal abrogations and annulments in defeasible logic. Logic Journal of IGPL **18**(1), 157–194 (2010)
42. Greco, S., Leone, N., Scarcello, F.: DATALOG with nested rules. In: Dix, J., Pereira, L.M., Przymusinski, T.C. (eds.) LPKR Workshop 1997. LNCS, vol. 1471, pp. 52–65. Springer (1997). https://doi.org/10.1007/BFb0054789
43. Grundy, J.: Transformational hierarchical reasoning. Computer Journal **39**(4), x3–302 (1996). https://doi.org/10.1093/comjnl/39.4.291
44. Hart, H.L.A.: The Concept of Law. Clarendon Press, Oxford (1994)
45. Horvitz, E.: Principles and applications of continual computation. Artificial Intelligence **126**(1-2), 159–196 (2001). https://doi.org/10.1016/S0004-3702(00)00082-5
46. Hovhannisyan, H., Djidjian, R.: Building a general theory of meta-argumentation. Metaphilosophy **48**(3), 345–354 (2017)
47. Kravari, K., Bassiliades, N.: A survey of agent platforms. Journal of Artificial Societies and Social Simulation **18**(1), 11 (2015). https://doi.org/10.18564/jasss.2661
48. Lewis, D.: Counterfactuals. Blackwell (1973)

49. Lifschitz, V., Tang, L., Turner, H.: Nested expressions in logic programs. Annals of Mathematics and Artificial Intelligence **25**(3-4), 369–389 (1999). https://doi.org/10.1023/a:1018978005636
50. Lloyd, J.W., Topor, R.W.: Making prolog more expressive. Journal of Logic Programming **1**(3), 225–240 (1984). https://doi.org/10.1016/0743-1066(84)90011-6
51. Lloyd, J., Topor, R.: Making prolog more expressive. The Journal of Logic Programming **1**(3), 225–240 (1984). https://doi.org/10.1016/0743-1066(84)90011-6
52. Makinson, D.: On a fundamental problem of deontic logic. In: McNamara, P., Prakken, H. (eds.) Norms, Logics and Information Systems. New Studies in Deontic Logic and Computer Science, Frontiers in Artificial Intelligence and Applications, vol. 49, pp. 29–53. IOS Press, Amsterdam (1999)
53. Makinson, D., van der Torre, L.: Permission from an input/output perspective. Journal of Philosophical Logic **32**(4), 391–416 (2003)
54. Makinson, D., Van Der Torre, L.: Input/output logics. Journal of philosophical logic **29**(4), 383–408 (2000)
55. Malerba, A., Rotolo, A., Governatori, G.: Interpretation across legal systems. In: Proc. JURIX 2016. IOS Press (2016)
56. Malerba, A., Rotolo, A., Governatori, G.: A logic for the interpretation of private international law. In: Rahman, S., Armgardt, M., Nordtveit Kvernenes, H.C. (eds.) New Developments in Legal Reasoning and Logic. Springer, Dordrecht (2021)
57. Mcdowell, R., Miller, D.: Reasoning with higher-order abstract syntax in a logical framework. ACM Transactions on Computational Logic **3**(1), 80–136 (2002). https://doi.org/10.1145/504077.504080
58. Modgil, S., Bench-Capon, T.: Metalevel argumentation. Journal of Logic and Computation **21**(6), 959–1003 (2011). https://doi.org/10.1093/logcom/exq054
59. Momigliano, A., Ambler, S.: Multi-level meta-reasoning with higher-order abstract syntax. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **2620**, 375–391 (2003)
60. Murray, N., Rosenthal, E.: Adapting classical inference techniques to multiple-valued logics using signed formulas. Fundamenta Informaticae **21**(3), 237–253 (1994). https://doi.org/10.3233/FI-1994-2135
61. Nute, D.: Defeasible logic. In: Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 3. Oxford University Press (1987)
62. Nute, D.: Topics in Conditional Logic. Reidel (1980)
63. Olivieri, F., Cristani, M., Governatori, G.: Compliant business processes with exclusive choices from agent specification. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **9387**, 603–612 (2015)
64. Olivieri, F., Governatori, G., Cristani, M., Sattar, A.: Computing defeasible metalogic. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **12678 LNAI**, 69–84 (2021)
65. Olivieri, F., Cristani, M., Governatori, G.: Compliant business processes with exclusive choices from agent specification. In: PRIMA 2015. LNCS, vol. 9387, pp. 603–612. Springer (2015). https://doi.org/10.1007/978-3-319-25524-8_43
66. Olivieri, F., Governatori, G., Cristani, M., Sattar, A.: Computing defeasible metalogic. In: Faber, W., Friedrich, G., Gebser, M., Morak, M. (eds.) Logics in Artificial Intelligence - 17th European Conference, JELIA 2021, Virtual Event, May 17-20, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12678, pp. 69–84. Springer (2021), https://doi.org/10.1007/978-3-030-75775-5_6

67. Olivieri, F., Governatori, G., Scannapieco, S., Cristani, M.: Compliant business process design by declarative specifications. In: PRIMA 2013. LNCS, vol. 8291, pp. 213–228. Springer (2013). https://doi.org/10.1007/978-3-642-44927-7_15
68. Papadias, D., Egenhofer, M.: Algorithms for hierarchical spatial reasoning. GeoInformatica **1**(3), 251–273 (1997). https://doi.org/10.1023/A:1009760430440
69. Paulson, L.: The foundation of a generic theorem prover. Journal of Automated Reasoning **5**(3), 363–397 (1989). https://doi.org/10.1007/BF00248324
70. Rotolo, A., Governatori, G., Sartor, G.: Logic and the law: Philosophical foundations, deontics, and defeasible reasoning. In: Gabbay, D., Horty, J., Parent, X., van der Meyden, R., van der Torre, L. (eds.) Handbook of Deontic Logic and Normative Systems, vol. 2. College Publications, London (2021)
71. Rowe, A., Watkins, P.: Beyond expert systems-reasoning, judgment, and wisdom. Expert Systems With Applications **4**(1), 1–10 (1992). https://doi.org/10.1016/0957-4174(92)90035-Q
72. Russell, S., Wefald, E.: Principles of metareasoning. Artificial Intelligence **49**(1-3), 361–395 (1991). https://doi.org/10.1016/0004-3702(91)90015-C
73. Sartor, G.: Legal Reasoning: A Cognitive Approach to the Law. Springer (2005)
74. Seidel, K., Morgan, C.: Hierarchical reasoning in probabilistic csp. Programming and Computer Software **23**(5), 239–250 (1997)
75. Song, I., Governatori, G.: Nested rules in defeasible logic. In: RuleML 2005 Conference, Galway, Ireland. LNCS, vol. 3791, pp. 204–208. Springer (2007)
76. Stolpe, A.: A theory of permission based on the notion of derogation. J. Applied Logic **8**(1), 97–113 (2010)
77. Stone, P.: EU Private International Law. Elgar European law, Edward Elgar (2014)
78. Tarski, A., Corcoran, J.: What are logical notions? History and Philosophy of Logic **7**(2), 143–154 (1986). https://doi.org/10.1080/01445348608837096
79. von Wright, G.H.: Norm and action: A logical inquiry. Routledge and Kegan Paul (1963)