# A DEONTIC ARGUMENTATION FRAMEWORK TOWARDS DOCTRINE REIFICATION

RÉGIS RIVERET
*CSIRO, Australia*
regis.riveret@data61.csiro.au

ANTONINO ROTOLO
*University of Bologna, Italy*
antonino.rotolo@unibo.it

GIOVANNI SARTOR
*European University Institute, Italy*
giovanni.sartor@eui.eu

**Abstract**

A modular rule-based argumentation system is proposed to represent and reason upon conditional norms featuring obligations, prohibitions, and (strong or weak) permissions. The approach is based on common constructs in computational models of argument: rule-based arguments, argumentation graphs, argument labelling semantics and statement labelling semantics. Deontic reasoning patterns are captured with defeasible rule schemata to the greatest extent, towards the reification of doctrinal pieces. We show then that bivalent statement labellings can fall short to address normative completeness, and for this reason, we propose to use trivalent labelling semantics. Given an argumentation graph, deontic statuses can be computed efficiently. The system is illustrated with a scenario featuring a violation and a contrary-to-duty obligation.

**Keywords:** Knowledge representation and reasoning, computational argumentation, legal reasoning, normative systems.

We would like to thank anonymous reviewers for their remarks, and Guido Governatori for discussions on the work reported here.

## 1 Introduction

There exist multiple formalisms to capture deontic reasoning [20, 35]. As deontic reasoning is often embedded into legal reasoning, and because legal reasoning is naturally formalised in formal models of arguments, deontic argumentation frameworks can be elegant and convenient formalisms to capture arguments in favour or against deontic statements. Yet, argumentation is more generally used to address defeasible claims raised on the basis of partial and conflicting pieces of information, that is, defeasible claims in uncertain contexts. Hence, argumentation may be well suited to reason upon deontic statements in uncertain contexts.

As interests in deontic systems can vary, diverse requirements or considerations can be put forward to build deontic argumentation frameworks. For example, some emulation of human discourses may be a key desideratum, while computational complexity may be a requisite for practical applications. For our purposes, we will pay attention to human interface (explainability, emulation and isomorphism), as well as more technical considerations (parsimony, modularity and computational efficiency), and a legit requirement regarding normative completeness. These design considerations are exposed in more detail later in the paper.

Bearing in mind specific design considerations, we explore a modular argumentation system. The argumentation system amounts to a knowledge-based system where conditional norms and common sense knowledge are represented as object rules into a knowledge base. The argumentation process then runs in three stages. In the first stage, arguments are built from object rules in the knowledge base by using some inference apparatus. In the second stage, arguments are labelled to reflect their acceptance. In the final stage, statements are labelled at their turn and normative completeness ensured.

To build such a knowledge-based system, we can roughly state that two main approaches can be taken: a knowledge-based (KB) approach and an inference-based (IB) approach. In the KB approach, maximum knowledge is expressed in the knowledge base, while the inference apparatus is restricted and not domain-specific. Possibly, inference schemata too are included in the knowledge base. Archetypes of the KB approach are Hilbert-style systems which usually code knowledge in axioms and axiom schemata with a few inference rules. In contrast, the IB approach tends to capture portions of knowledge in the inference apparatus which handles the knowledge base. Examples of the IB approach are Gentzen-style systems where the number of inference rules is typically larger than in Hilbert-style systems. Thus, we can say that in general the KB approach limits the number of inference patterns, while the IB approach favours a diversity of inference patterns.

Both approaches have strengths and weaknesses. If something cannot be achieved with the KB approach (e.g. the pursuit of some natural reasoning patterns, specific inferences or features such as some quantitative assessment, or a certain computational complexity), then one can take the IB approach and attempt to devise some specific inference patterns. Some trade-off may be obtained, for example by devising systems with an 'inference knowledge base' storing a diversity of inference patterns. This approach may be called here the inference-knowledge-based (IKB) approach. The advantage of the IKB approach in automated systems is that designers and users can have an easy and direct access to reasoning patterns which would be otherwise hidden in the implemented inference machinery.

Any of these options can be adopted to build a computational system where norms are represented as rules from which arguments are formed to decide about the status of deontic statements. Within the IB approach, deontic reasoning is captured with specific inference patterns (e.g. deontic detachment). Examples of such an approach in deontic argumentation are [7, 47]. Within the KB approach, deontic reasoning is written as object rules in the knowledge base with no specific inference patterns. For example, a weak permission can be supported by some arguments built out from object rules (instead of hard-coding its support from the absence of prohibitions). These object rules may be seen as a reification of inference rules into the language of the knowledge base, but they are not inference rules because they are part of the object knowledge base. If the inference rules for deontic reasoning are part of an inference knowledge base, then we have a deontic IKB approach. An example of the IKB approach is [48] which uses ASPIC$^+$ framework [37] and where inference rules are tuned for deontic reasoning.

In this paper, we explore a hybrid approach: first, we use defeasible rule schemata to account for deontic reasoning, towards doctrine reification. In particular, we investigate the reification of the principle of prohibition to address normative completeness. In that sense, we adopt a KB approach. However, for aspects of normative completeness we could not cover with defeasible rule schemata, we exploit the inference apparatus through labelling semantics, and thus we complete our KB approach with labelling inferences.

**Contribution.** A modular rule-based argumentation system is proposed to capture normative knowledge and reason upon it. To do so, we adopt a KB approach to the greatest extent through defeasible rule schemata, towards doctrine reification. In particular, we explore the reification of the principle of prohibition to address normative completeness. Then, for aspects of normative completeness not covered by defeasible rules schemata, we exploit the inference apparatus through labelling semantics. The system is actually constructed on quite common definitions of rule-based arguments, argumentation graphs, argument and statement labelling semantics. By doing so, we aim at showing that deontic argumentation frameworks can be devised on the basis of common argumentation constructs, with little modifications to standard inference apparatus. As a consequence, the system inherits the modularity of labelling argumentation systems, as it can be tuned to capture diverse refinements on argument and statement labellings; each refinement variant leading to a specific deontic argumentation framework. In that regard, we will see that standard bivalent statement labellings are not sufficient to achieve normative completeness, and for this reason, we propose to use a trivalent labelling semantics. Through this journey, we will retrieve three possible interpretations of the principle of prohibition.

**Outline.** In Section 2, some design considerations are exposed to build the computational system. In Section 3, we present a simple argumentation setting which is our basis of our deontic argumentation system. In Section 4, deontic knowledge is captured in deontic defeasible theories, from which argumentation graphs are produced, and deontic statements are labelled. Section 5 illustrates the overall approach. Section 6 evaluates the system with respect to the elicited design considerations. We discuss related work in Section 7, before concluding.

## 2 Some Design Considerations

Some design considerations may be useful to guide the construction of deontic argumentation systems. We will develop our system by bearing in mind 'human interface' and 'inference' issues, and a 'legit' requirement concerning normative completeness.

Interface considerations regard human-centric requirements to facilitate user interactions with the system, and we will value explainability, emulation, and isomorphism.

**Explainability** Computation of normative accounts should be explainable. Yet, the question 'What is a good explanation?' is quite elusive and goes far beyond the present work, see diverse conceptions in philosophy e.g. [5, 26, 31] or psychology [28, 32]. Here we focus on a specific aspect of explainability, namely, the explanation of why a certain statement is accepted or rejected, in the context of all relevant arguments. Thus, explainability means for our purposes that the output acceptance labelling of statements (i.e. conclusions) should be easily explainable to humans (-in-the-loop) by the interplay of arguments.

**Emulation** Computational models of argument are often inspired by argumentation as practised by humans. Following this line, argumentation for deontic reasoning should somehow emulate the way humans argue about deontic statements. In particular, we would like to account for full-fledged arguments built from doctrinal pieces, for example those arguments supporting permissions which are implicit in normative systems.

**Isomorphism** A well-established principle to build knowledge-based systens in the legal domain states that there should be an 'isomorphic' correspondence between the knowledge base and the sources [8]. Isomorphism can facilitate the development, verification, validation and maintenance of the knowledge base, and the provision of more intelligible explanations to end-users. Isomorphism implies here that we have constructs to account for conditional norms featuring obligations, prohibitions and permissions.

Inference considerations are more technical, and we will focus on parsimony, modularity and efficiency.

**Parsimony** As various deontic argumentation formalisms can be proposed to meet our needs, we shall prefer the simplest or most parsimonious formalisms, that is formalisms where unnecessary elements and constructions are excluded. Beyond formal elegance, a parsimonious deontic formalism is important because of its greater falsifiability, and because it can ease integration with other endeavours in argumentation, such as e.g. judgement aggregation, probabilistic, strategic or machine learning undertakings [9, 40, 41, 43]. Parsimony can be analysed in various ways – we will appreciate it at the level of inference machinery and the knowledge base.
**Modularity** To facilitate semantics variants, verification, validation and maintenance of eventual 'argument-based software systems', the deontic argumentation system may be, so to say, modular by design. This can be achieved, for example, by developing a module for each labelling stage, such that every module can be tuned. In this view, we prefer to investigate a deontic argumentation system from which various deontic argumentation frameworks can be drawn.
**Efficiency** Concerning computational complexity, one may seek for systems which can operate efficiently for practical ends, and thus argumentation semantics which can be accompanied with efficient algorithms. Following computational complexity theory, an algorithm is efficient if it can be performed in polynomial time.

Finally, beyond consistency according to which incompatible (deontic) statements should not be accepted together, the legit requirement regards the normative completeness of our formal system.

**Completeness** We are after deontic argumentation frameworks which are complete. Prescriptions may be such that there exist 'normative gaps' (or 'legal gaps' in legal systems), i.e., in some cases, something is neither explicitly obligatory, permitted nor prohibited. Normative completeness refers here to the completeness quality of the deontic argumentation system: a system is complete if, and only if, anything is eventually obligatory, permitted or prohibited, even though, for example, something is not regulated by any primary norms. To address normative completeness, the 'principle of prohibition', according to which everything that is not prohibited is permitted, can be put forward, and we will do so. Yet, such a principle can be interpreted in various ways, see e.g. [1], which should be accounted for.

Many argumentation settings in the literature cater for various human interface and inference aspects in a way or another. We are thus greatly inspired from these systems to build our deontic argumentation system in the next sections. However, normative completeness has attracted less attention. and we shall thus pay particular attention to it. Above-mentioned design considerations will be used to evaluate the system later in Section 6.

# 3 Argumentation System

This section presents a lightweight ASPIC$^+$-like argumentation system along with a common sequential model consisting of the following stages [4]: definition of the language, defeasible theories and argumentation graph production, argument acceptance/justification and statement acceptance/justification. These stages are developed in the remainder of the section.

## 3.1 Language

Building blocks of the formalism are so-called 'literal statements' (which are later further specified to cater for deontic modalities).

**Definition 3.1.** *A* ***literal statement*** *is either a plain literal statement or a modal literal statement, where*
- *a* ***plain literal statement*** *is either an atomic proposition $p$ or the negation of an atomic proposition, i.e. $\neg p$, and*
- *a* ***modal literal statement*** *is a statement of the form $\Box\gamma$ or $\neg\Box\gamma$, such that $\Box$ is a placeholder for any modal operator and $\gamma$ is a plain literal statement.*

**Notation 3.1.** *For any plain literal statement $\gamma$, its complement is written $\overline{\gamma}$. Hence, if $\gamma$ is $p$ then $\overline{\gamma}$ is $\neg p$, and if $\gamma$ is $\neg p$ then $\overline{\gamma}$ is $p$.*

Literal statements can be put in relation through defeasible rules. Defeasible rules represent conditionals of the form '*if ... then ... unless ...*'. For the sake of simplicity, we deal with defeasible rules only, i.e. rules that can be defeated by other rules.

**Definition 3.2.** *A* ***defeasible rule*** *over a set of literal statements $\mathscr{S}$ is a construct of the form $r : \varphi_1, \ldots, \varphi_n, \sim \varphi'_1, \ldots, \sim \varphi'_m \Rightarrow \varphi$ with $0 \le n$ and $0 \le m$, and where*
- *$r$ is the unique identifier of the rule, and*
- *for any $0 \le i \le n$ and $0 \le j \le m$, $\varphi_i, \varphi'_j, \varphi \in \mathscr{S}$ are all literal statements.*

Given a rule as in Definition 3.2, the set $\{\varphi_1, \ldots, \varphi_n, \sim \varphi'_1, \ldots, \sim \varphi'_m\}$ is the body of the rule. The singleton $\{\varphi\}$ is the head of the rule.

**Notation 3.2.** *Given a rule $r$ as in Definition 3.2,*
- *the body of $r$ is denoted $\mathrm{Body}(r)$, i.e. $\mathrm{Body}(r) = \{\varphi_1, \ldots, \varphi_n, \sim \varphi'_1, \ldots, \sim \varphi'_m\}$,*
- *the head of $r$ is denoted $\mathrm{Head}(r)$, i.e. $\mathrm{Head}(r) = \{\varphi\}$,*
- *the set of propositions of $r$ is denoted $\mathrm{Prop}(r)$, i.e. $\mathrm{Prop}(r) = \{p \mid p, \neg p, \sim p, \sim \neg p, \Box p, \neg\Box p, \Box\neg p, \neg\Box\neg p, \sim \Box p, \sim \neg\Box p, \sim \Box\neg p, \sim \neg\Box\neg p \in \mathrm{Body}(r) \cup \mathrm{Head}(r)\}$.*

*The set of propositions of a set of rules $Rules$ is denoted $\mathrm{Prop}(Rules)$, i.e. $\mathrm{Prop}(Rules) = \bigcup_{r \in Rules} \mathrm{Prop}(r)$.*

A defeasible rule $r : \varphi_1, \ldots, \varphi_n, \sim \varphi'_1, \ldots, \sim \varphi'_m \Rightarrow \varphi$ can be roughly read as follows: '*if* $\varphi_1$ *and ... and* $\varphi_n$ *are supported then* $\varphi$ *is defeasibly supported, unless* $\varphi'_1$ *is supported or ... or unless* $\varphi'_m$ *is supported*'. We specify later what 'supported' means here. The symbol $\sim$ can be viewed as a sort of negation as failure, but it may be rather understood as a point of attack (as we will see soon) to avoid any confusion with formal semantics from the literature on the concept of negation as failure.

Rules may head to incompatible statements. Incompatibilities amongst statements are captured in a conflict relation defined as a binary relation over literal statements.

**Definition 3.3.** *A* ***conflict relation*** *'Conflicts' over a set of literal statements* $\mathscr{S}$ *is a binary relation over* $\mathscr{S}$*, i.e.* $Conflicts \subseteq \mathscr{S} \times \mathscr{S}$.

**Notation 3.3.** *The set propositions of a conflict relation 'Conflicts' is denoted* $\mathrm{Prop}(Conflicts)$*, i.e.* $\mathrm{Prop}(Conflicts) = \{p \mid (\varphi, \varphi') \in Conflicts : \varphi = p, \neg p, \Box p, \neg\Box p, \Box\neg p, \neg\Box\neg p, \text{ or } \varphi' = p, \neg p, \Box p, \neg\Box p, \Box\neg p, \neg\Box\neg p\}$.

The relation is meant to be 'well-formed'. For example, we may constrain conflicts such that for any literal statement $\gamma$ and its complement $\overline{\gamma}$ we have $Conflicts(\gamma, \overline{\gamma})$. Well-formedness is left unspecified at this stage, it will be specified in the deontic development of the relation. We may also further refine the relation with asymmetric and symmetric conflicts to deal with contrary or contradictory statements, but such sophistications are not necessary for our ends.

When two rules have conflicting heads, one rule may prevail over another one. To possibly disentangle such cases, we use a superiority relation $\succ$ over rules. Informally, $s \succ r$ states that rule $s$ prevails over rule $r$.

**Definition 3.4.** *A* ***superiority relation*** $\succ$ *over a set of rules Rules is an antireflexive and antisymmetric binary relation over Rules, i.e.* $\succ \subseteq Rules \times Rules$.

As the superiority relation is antireflexive and antisymmetric, for any rule $r$ it does not hold that $r \succ r$, and for two distinct rules $r$ and $r'$ we cannot have both $r \succ r'$ and $r' \succ r$.

## 3.2 Defeasible theories and argumentation graphs

A defeasible theory lists a set of rules, a conflict relation and a superiority relation.

**Definition 3.5.** *A* ***defeasible theory*** *is a tuple* $\langle Rules, Conflicts, \succ \rangle$ *where*

- *Rules is a set of rules, and*
- *Conflicts is a conflict relation, and*
- $\succ$ *is a superiority relation over Rules.*

**Notation 3.4.** *Given a defeasible theory $T = \langle Rules, Conflicts, \succ \rangle$,*

- *the set of rules Rules, the relation Conflicts, and the relation $\succ$ are denoted* $\mathrm{Rules}(T)$*, Conflicts*$(T)$*, and* $\succ (T)$ *respectively,*
- *the set propositions of $T$ is denoted* $\mathrm{Prop}(T)$*, i.e.* $\mathrm{Prop}(T) = \mathrm{Prop}(Rules) \cup \mathrm{Prop}(Conflicts)$.

By chaining rules of a defeasible theory, we can build arguments. Arguments are captured by the following definition, which is much inspired from other rule-based argumentation frameworks as exposed for example in [11, 37].

**Definition 3.6.** *An **argument** $A$ constructed from a defeasible theory $\langle Rules, Conflicts, \succ \rangle$ is a finite construct of the form: $A : A_1, \ldots A_n, \sim \varphi_1, \ldots, \sim \varphi_m \Rightarrow_r \varphi$ with $0 \leq n$ and $0 \leq m$ and such that*

- *$A$ is the unique identifier of the argument, and*
- *$A_1, \ldots, A_n$ are arguments constructed from the defeasible theory $\langle Rules, Conflicts, \succ \rangle$, and*
- *$\varphi$ is the conclusion of argument $A$; the conclusion of an argument $A$ is denoted* $\mathrm{conc}(A)$*, i.e.* $\mathrm{conc}(A) = \varphi$*, and*
- *there exists a rule $r \in Rules$ such that* $r : \mathrm{conc}(A_1), \ldots, \mathrm{conc}(A_n), \sim \varphi_1, \ldots, \sim \varphi_m \Rightarrow \varphi$.

**Definition 3.7.** *Given an argument $A : A_1, \ldots A_n, \sim \varphi_1, \ldots, \sim \varphi_m \Rightarrow_r \varphi$, the set of its subarguments* $\mathrm{Sub}(A)$*, the set of its direct subarguments* $\mathrm{DirectSub}(A)$*, the last inference rule* $\mathrm{TopRule}(A)$*, and the set of all the rules in the argument* $\mathrm{Rules}(A)$ *are defined as follows:*

- $\mathrm{Sub}(A) = \mathrm{Sub}(A_1) \cup \ldots \cup \mathrm{Sub}(A_n) \cup \{A\}$,
- $\mathrm{DirectSub}(A) = \{A_1, \ldots, A_n\}$,
- $\mathrm{TopRule}(A) = (r : \mathrm{conc}(A_1), \ldots, \mathrm{conc}(A_n), \sim \varphi_1, \ldots, \sim \varphi_m \Rightarrow \varphi)$,
- $\mathrm{Rules}(A) = \mathrm{Rules}(A_1) \cup \ldots \cup \mathrm{Rules}(A_n) \cup \{\mathrm{TopRule}(A)\}$.

According to Definition 3.6, an argument without direct subarguments has thus the form $A : \sim \varphi_1, \ldots, \sim \varphi_m \Rightarrow_r \varphi$ with $0 \leq m$. Arguments may be infinite, and we may have an infinite set of arguments constructed from a defeasible theory, however, we will work with finite sets of finite arguments.

Last item in Definition 3.6 asserts that arguments are built using one single implicit inference rule, namely (defeasible) modus ponens of the form

$$\frac{\begin{array}{l} r : \varphi_1, \ldots, \varphi_n, \sim \varphi'_1, \ldots, \sim \varphi'_m \Rightarrow \varphi \\ \quad \varphi_1, \ldots, \varphi_n \end{array}}{\varphi}$$

By using this simple and single inference rule pertaining to common sense reasoning, our intention is to build arguments through reasoning steps which come ‘as close as possible to actual reasoning'. Yet, by contrast to Gentzen-style calculi (sequent calculus and natural deduction) which have a minimal number of axioms and multiple inference rules, we propose to have one single inference rule, namely a defeasible modus ponens to meet our parsimony requirement. If one wants to bring the framework closer to actual reasoning, then the framework can be certainly developed in that direction by including sundry inference rules to build arguments.

Arguments may conflict and thus attacks between arguments may appear. We reckon two types of attacks: rebuttals (clash of incompatible conclusions) and undercuttings[1] (attacks on negation as failure premises). In regard to rebuttals, we assume a preference relation over arguments determining whether two rebutting arguments mutually attack each other or only one of them (being preferred) attacks the other. The preference relation over arguments can be defined in various ways on the basis of the preference over rules. We adopt a simple last-link ordering (back to e.g. [38]), according to which an argument $A$ is preferred over another argument $B$, denoted as $A \succ B$, if, and only if, the rule $\mathrm{TopRule}(A)$ is superior to the rule $\mathrm{TopRule}(B)$, i.e. $\mathrm{TopRule}(A) \succ \mathrm{TopRule}(B)$. This leads us to adopt the following definition of attack relation, cf. other formulations e.g. in [37].

**Definition 3.8.** *An **attack relation** $\rightsquigarrow$ over a set of arguments $\mathscr{A}$ is a binary relation over $\mathscr{A}$, i.e. $\rightsquigarrow \subseteq \mathscr{A} \times \mathscr{A}$. An argument B attacks an argument A, i.e. $B \rightsquigarrow A$, iff B rebuts or undercuts A, where*

- *B rebuts A (on $A'$) iff $\exists A' \in \mathrm{Sub}(A)$ such that $\mathrm{conc}(B)$ and $\mathrm{conc}(A')$ are in conflict, i.e. $\mathit{Conflicts}(\mathrm{conc}(B), \mathrm{conc}(A'))$, and $A' \not\succ B$;*
- *B undercuts A (on $A'$) iff $\exists A' \in \mathrm{Sub}(A)$ such that $\sim \mathrm{conc}(B)$ belongs to the body of $\mathrm{TopRule}(A')$, i.e. $(\sim \mathrm{conc}(B)) \in \mathrm{Body}(\mathrm{TopRule}(A'))$.*

Arguments and attack relations can be then captured in Dung's abstract argumentation graphs, originally called abstract argumentation frameworks in [16].

**Definition 3.9.** *An **argumentation graph** is a pair $\langle \mathscr{A}, \rightsquigarrow \rangle$ where $\mathscr{A}$ is a set of arguments, and $\rightsquigarrow \subseteq \mathscr{A} \times \mathscr{A}$ is a binary relation of attack.*

**Notation 3.5.** *Given an argumentation graph $G = \langle \mathscr{A}, \rightsquigarrow \rangle$, we may denote $\mathscr{A}$ as $\mathscr{A}_G$ and $\rightsquigarrow$ as $\rightsquigarrow_G$.*

**Definition 3.10.** *An argumentation graph $\langle \mathscr{A}, \rightsquigarrow \rangle$ is an **argumentation graph constructed from a defeasible theory** iff $\mathscr{A}$ is the set of all arguments constructed from the defeasible theory.*

[1] The term undercutting is overloaded in argumentation literature and is used with different meanings in different contexts, cf. [37].

Clearly, the number of arguments in an argumentation graph constructed from a defeasible theory may not be polynomial in the number of rules of the theory. As complexity matters, we may focus on those theories from which argumentation graphs can be constructed efficiently. However, we may also be given an argumentation graph, and check its 'well-formedness', presumably against some theory. Anyhow, given an argumentation graph built (possibly efficiently) from a defeasible theory, we can then label arguments following argument labelling semantics.

### 3.3 Labelling semantics

Given an argumentation graph, the sets of arguments that are accepted or rejected, that is, those arguments that will survive or not to possible attacks, are computed using some semantics. For our purposes, we resort to labelling semantics as reviewed in [2,3]. Accordingly, we endorse $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labellings where each argument is associated with one label which is either $\mathsf{IN}$, $\mathsf{OUT}$, or $\mathsf{UND}$, respectively meaning that the argument is accepted, rejected, or undecided.

**Definition 3.11.** *A* $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$***-labelling*** *of an argumentation graph G is a total function* $L : \mathscr{A}_G \to \{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$.

**Notation 3.6.** *A* $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$*-labelling L may be represented as a tuple* $\langle \mathsf{IN}(L), \mathsf{OUT}(L), \mathsf{UND}(L) \rangle$ *where* $\mathsf{IN}(L)$ *stands for* $\{A \mid L(A) = \mathsf{IN}\}$*,* $\mathsf{OUT}(L)$ *for* $\{A \mid L(A) = \mathsf{OUT}\}$*, and* $\mathsf{UND}(L)$ *for* $\{A \mid L(A) = \mathsf{UND}\}$.

Most argument labellings studied in the literature are *complete labellings* [2]. An argumentation graph may have several complete $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labellings, we will focus on the unique complete labelling with the smallest set of labels $\mathsf{IN}$, namely the grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling.

**Definition 3.12.** *Let G denote an argumentation graph. A* ***complete*** $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$***-labelling*** *of G is a* $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$*-labelling such that for every argument A in* $\mathscr{A}_G$*:*
- *A is labelled* $\mathsf{IN}$ *iff all attackers of A are labelled* $\mathsf{OUT}$*,*
- *A is labelled* $\mathsf{OUT}$ *iff A has an attacker labelled* $\mathsf{IN}$.

**Definition 3.13.** *A complete* $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$*-labelling L is a* ***grounded*** $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$***-labelling*** *of an argumentation graph G if* $\mathsf{IN}(L)$ *is minimal (w.r.t. set inclusion) amongst all complete* $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$*-labellings of G.*

Since complete or grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labellings are total functions, if an argument is not labelled $\mathsf{IN}$ or $\mathsf{OUT}$, then it is labelled $\mathsf{UND}$.

The reason to focus on the grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling is that it is unique and it can be computed in a polynomial time, using e.g. Algorithm 1 [36]. The algorithm begins by labelling IN all arguments not being attacked or whose attackers are OUT (line 4), and then it iteratively labels OUT any argument attacked by an argument labelled IN (line 5). The iteration continues until no more arguments can be labelled IN or OUT (line 6); and if the argumentation graph is finite, then it terminates by labelling UND unlabelled arguments (line 7).

**Algorithm 1** Computation of a grounded {IN, OUT, UND}-labelling.

1: **input** An argumentation graph $G$,
2: $L_0 = (\emptyset, \emptyset, \emptyset)$,
3: **repeat**
4: $\quad \mathsf{IN}(L_{i+1}) \leftarrow \mathsf{IN}(L_i) \cup \{A \mid A \in \mathscr{A}_G$ is not labelled in $L_i$, and $\forall B \in \mathscr{A}_G$ : if $B$ attacks $A$ then $B \in \mathsf{OUT}(L_i)\}$
5: $\quad \mathsf{OUT}(L_{i+1}) \leftarrow \mathsf{OUT}(L_i) \cup \{A \mid A \in \mathscr{A}_G$ is not labelled in $L_i$, and $\exists B \in \mathscr{A}_G$ : $B$ attacks $A$ and $B \in \mathsf{IN}(L_{i+1})\}$
6: **until** $L_i = L_{i+1}$
7: **return** $(\mathsf{IN}(L_i), \mathsf{OUT}(L_i), \mathscr{A}_G \backslash (\mathsf{IN}(L_i) \cup \mathsf{OUT}(L_i)))$

Algorithm 1 is given here to show that the computation of the grounded labelling of an argumentation graph can be performed in polynomial time. It does not pre-empt the use of more efficient algortihms, see e.g. [14, 15].

Given a set of statements, a labelling of this set is a (preferably total) function associating any statement with a label. Different specifications for statement labellings are possible, see e.g. [4] where statement *acceptance* labellings are distinguished from statement *justification* labellings. For our purposes, we will work with acceptance labellings, and we first turn to the acceptance labelling semantics which is perhaps the simplest in a meaningful way, namely bivalent labelling semantics, according to which a statement is either accepted or not, without further sophistication. If a statement is accepted then it is labelled 'in', otherwise it is labelled 'ni'. As statements are labelled relatively to so-called argument acceptance labelling semantics, we have acceptance bivalent {in, ni}-labellings, but we will simply call them bivalent {in, ni}-labellings.

**Definition 3.14.** *Let $\mathfrak{L}$ be a set of* {IN, OUT, UND}*-labellings, $\mathscr{S}$ a set of literal statements. A **bivalent*** {in, ni}***-labelling** of $\mathscr{S}$ and from $\mathfrak{L}$ is a total function $K : \mathfrak{L}, \mathscr{S} \rightarrow \{\mathsf{in}, \mathsf{ni}\}$ such that for any argument labelling $L \in \mathfrak{L}$ and $\varphi \in \mathscr{S}$:*

- $K(L, \varphi) = \mathsf{in}$ *iff* $\exists A \in \mathsf{IN}(L) : \mathrm{conc}(A) = \varphi$,
- $K(L, \varphi) = \mathsf{ni}$ *otherwise.*

We can also take statement labellings which better exploit statuses of {IN, OUT, UND}-labellings [4]. For our very investigation into deontic argumentation, we consider trivalent

$\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labellings which reckon undecided statements.

**Definition 3.15.** *Let $\mathfrak{L}$ be a set of $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labellings, $\mathscr{S}$ a set of literal statements. A **trivalent** $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$**-labelling** of $\mathscr{S}$ and from $\mathfrak{L}$ is a total function $K : \mathfrak{L}, \mathscr{S} \rightarrow \{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$ such that for any argument labelling $L \in \mathfrak{L}$ and $\varphi \in \mathscr{S}$:*

- $K(L, \varphi) = \mathsf{in}$ *iff* $\exists A \in \mathsf{IN}(L) : \mathrm{conc}(A) = \varphi$, *and*
- $K(L, \varphi) = \mathsf{und}$ *iff* $\exists A \in \mathsf{UND}(L) : \mathrm{conc}(A) = \varphi$ *and* $\nexists A \in \mathsf{IN}(L) : \mathrm{conc}(A) = \varphi$, *and*
- $K(L, \varphi) = \mathsf{niund}$ *otherwise.*

**Notation 3.7.** *A bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labelling or a trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling $K$ may be represented as a tuple $\langle \mathsf{in}(K), \mathsf{ni}(K) \rangle$ and $\langle \mathsf{in}(K), \mathsf{und}(K), \mathsf{niund}(K) \rangle$ respectively, with the obvious meaning.*

For the sake of simplicity, since we will work with grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling semantics and because every argumentation graph has a unique grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling, we assume that any statement justification labelling simply corresponds to a statement acceptance labelling which can be a bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labelling or a trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling. The distinction between bivalent and trivalent labellings is later exploited in our deontic setting, in particular to address normative completeness.

# 4 Deontic Argumentation System

Having laid out a simple rule-based argumentation system, we can now specify a deontic version of it. To do so, we first specify our deontic statements, and then adopted labelling semantics are discussed.

## 4.1 Deontic language

Legal and deontic reasoning expose varied concepts – ranging from basic obligations and permissions to liberties and rights. For our purposes, we focus on basic concepts in deontic reasoning, namely obligations, prohibitions, and permissions.

Obligations are at the core of our deontic system, and prohibitions are viewed as a by-product of obligations: ‘something is prohibited’ is equivalently expressed by stating that its opposite is obligatory. Permissions can be understood in terms of obligations too: a permission for something expresses that the opposite is not obligatory.

Accordingly, and for the sake of simplicity, the attention is restricted to a propositional language which is supplemented with a single deontic operator $\mathsf{O}$ which indicates an obligation. Hence, we assume a language $\mathscr{L}_D$ whose literal statements are defined as follows, cf. Definition 3.1.

**Definition 4.1.** *A literal statement of a language $\mathscr{L}_D$ is either a plain literal statement or a deontic literal statement, where:*

- *a **plain literal statement** is either an atomic proposition $p$ or the negation of an atomic proposition, i.e. $\neg p$, and*
- *a **deontic literal statement** is a statement of the form $\mathsf{O}\gamma$ or $\neg\mathsf{O}\gamma$ such that $\gamma$ is a plain literal statement.*

Prohibitions and permissions are captured by assuming that a prohibition $\mathsf{F}\gamma$ is equivalently expressed by the obligation $\mathsf{O}\overline{\gamma}$, and a permission $\mathsf{P}\gamma$ is syntactically equivalent to $\neg\mathsf{O}\overline{\gamma}$.

**Notation 4.1.** *As syntactic sugar, we may write $\mathsf{O}\overline{\gamma}$ as $\mathsf{F}\gamma$, and $\neg\mathsf{O}\overline{\gamma}$ as $\mathsf{P}\gamma$ (and vice versa). Accordingly, $\mathsf{O}p$ stands for $\mathsf{F}\neg p$, $\mathsf{O}\neg p$ for $\mathsf{F}p$, $\mathsf{P}p$ for $\neg\mathsf{O}\neg p$, and $\mathsf{P}\neg p$ for $\neg\mathsf{O}p$.*

Anything is meant to be either obligatory, prohibited or permitted. However, in practice, and as remarked by legal scholars, there can be gaps (see e.g. [1]), i.e. in some cases, something is neither obligatory, permitted nor prohibited. Normative completeness refers here to the completeness quality of the deontic argumentation system: a system is complete if, and only if, anything is eventually obligatory, permitted nor prohibited, even though, for example, something is not regulated by any primary norms.

To address normative completeness, we will exploit the well-established principle of prohibition, which can be formulated as follows: 'everything that is not prohibited is permitted'. The principle is shared by various legal systems (for instance it corresponds to the 'norma generale exclusiva' or 'principio di libertá' in Italian system); however, we have to note that the principle does not apply in all legal systems.

In criminal law for example (in civil law systems), if being permitted to do an action means not being subject to sanction for it, then there is the idea that every criminal sanction must be explicitly stated by positive law (the sanction cannot be obtained by analogy). This principle, according to European Court of Human Rights (case Ozturk v. Germany, 1984), goes beyond criminal law strictly understood, applying to any sanction having punitive character, namely, going beyond compensation of damage (e.g. administative fines). Therefore if an action is not punished by a norm (it is not prohibited), one can conclude that the legal system is committed not to punish (the judge should not punish it), so that the action is definitely permitted, as long as the explicit rules remain the same (the law says implicitly that the action is subject to no sanction) according to criminal law. The idea can be retrieved in the principle 'nullum crimen sine lege' (no crime without law), i.e. everything that is not explicitly prohibited should be considered as permitted.

In private law, on the contrary, it is possible for the judge to establish a sanction (compensation for damages) using analogy or other legal constructions even for actions that are not explicitly prohibited by the law. In civil law, the fact that no norm explicitly establishes a sanction for an action does not ensure that the action will not be sanctioned, there is just a gap: we do not know what will happen. For instance, in many legal systems, before consumer protection laws were enacted, judges started to condemn producers to compensate damages caused by defective products. This required overcoming, in the absence of an explicit rule, the idea that producers owned no duty of care to consumers.

Hence, in general, the principle of prohibition and our upcoming account of it apply to normative systems where it makes sense to use it.

Besides, the principle can be diversely interpreted. In a first interpretation, the principle can be read to stress a mere tautology: no prohibition $(\neg \mathsf{O}\overline{\gamma})$ is equivalent to a permission $(\mathsf{P}\gamma)$. Such a tautology can be, for example, syntactically captured by writing $\neg \mathsf{O}\overline{\gamma}$ as $\mathsf{P}\gamma$ and vice versa (as we do in Notation 4.1), but this will not appear to successfully fill any gaps in our formal framework. In a second interpretation, we can adopt the reading according to which a thing is permitted unless it is prohibited. Following this interpretation the principle of prohibition is not a tautology anymore, but rather a normative principle included in the normative system being considered, to effectively fill gaps by producing permissions.

In this context, we may distinguish strong and weak permissions (similarly as notably retained by G. H. von Wright [49]), where a strong permission is a permission derived from permissive norms, and a weak permission for $\varphi$ is a permission which is accepted if the prohibition of or on $\varphi$ is not accepted. This conception makes reference, for example, to C.E. Alchourrón and E. Bulygin who state: ‘Weak permission differs from strong permission in an important way: strong permission expresses a positive fact (the existence of a permissive norm), whereas weak permission refers to a negative fact: the non-existence of a prohibitive norm’ [1], see also [33,45,46]. In this view, a strong permission does not fully correspond to what we may call an ‘explicit permission’ (i.e. a permission which is explicitly formulated in a permissive norm): for example, if one derives in the system the acceptance of $\mathsf{P}p$ and $\mathsf{P}q$, one might infer $\mathsf{P}(p \text{ and } q)$, which is not explicit in the sense that it is derived, but is nevertheless strong. The strengths of permissions are not exclusive: we can have a strong and weak permission for the same thing, the two permissions would not be incompatible. In our framework, the strengths of permissions are not directly represented in the language, essentially because laypersons or jurists do not usually specify the strength of permissions in their discourses. In this view, a weak or a strong permission is seen as a permission tout court.

A defeasible rule can specify varied relationships amongst (deontic) literal statements of a given language $\mathscr{L}_D$. Such rules are called normative defeasible rules.

**Definition 4.2.** *Given a language $\mathscr{L}_D$, a **normative defeasible rule** is a defeasible rule over a set of literal statements in $\mathscr{L}_D$.*

Normative rules are partitioned into *foreground rules* and *background rules*. Foreground rules provide substantive legal regulations, for particular domains of the law, while background rules express deontic assumptions underlying the normative system being dealt with.

Foreground rules are domain-dependent rules. They are meant here to represent primary norms, and thus they can be classified as either *constitutive rules* or *regulative rules*. The effect of a constitutive rule is to define a term as understood in a given situation or to 'create' an institutional entity from a set of brute or institutional facts. A regulative rule, on the other hand, triggers a 'deontic' effect (obligation, prohibition, permission) when certain conditions are established. While constitutive and regulative norms have been formally approached in various (and sometimes sophisticated) ways in the literature [20], the distinction is simply addressed in the present system: the consequent of the rule is a plain literal for constitutive rules, and a deontic literal for regulative rules. In that regard, we can note that a regulative rule heading to a (strong) permission would typically specify an exception to an obligation, as notably discussed by A. Ross [44], but such a rule can also be used to stress a permission and clarify its conditions.

Background rules are domain-independent. They underpin core deontic reasoning. These background rules can be viewed as defeasible rule schemata which are isomorphic to some pieces of (possibly very basic) legal doctrines. Instead of giving a formal definition of background defeasible rules, we give here some examples of such schemata:

$\mathsf{d\_}\gamma : \mathsf{O}\gamma \Rightarrow \mathsf{P}\gamma$ An obligation $\mathsf{O}\gamma$ implies a permission $\mathsf{P}\gamma$ (cf. Axiom 'D' in deontic logics).

$\mathsf{p\_}\gamma : \Rightarrow \mathsf{P}\gamma$ Anything is permitted prima facie.

$\mathsf{k\_}\gamma : \sim \mathsf{O}\overline{\gamma} \Rightarrow \mathsf{P}\gamma$ Anything that is not prohibited is permitted.

These background rules can be essentially employed to build arguments supporting permissions. In particular, the second and third rules can be used to derive permissions even if there are no applicable foreground permissive rules heading to such permissions. As to the terminology, although such permissions are the consequent of some rules, we may say for now that these background rules head to weak permissions because these rules are meant to ensure that a permission can be accepted if any contrary prohibition is non-existent or rejected, and even though there is no foreground permissive rule heading to such a permission (as we will see later).

Given some background rules, sets of background rules can be formed, possibly to account for various doctrinal systems. In our case, and for our purposes, we will work with the sets $\{\mathsf{d\_}\gamma, \mathsf{p\_}\gamma\}$ and $\{\mathsf{d\_}\gamma, \mathsf{k\_}\gamma\}$.

**Definition 4.3.** *A set of background defeasible rule schemata B is*

- *a* ***permissive by default set of background defeasible rule schemata*** *iff* $B = \{\mathsf{d\_}\gamma, \mathsf{p\_}\gamma\}$*;*

- *a **Kelsenian permissive set of background defeasible rule schemata** iff $B = \{\mathsf{d\_}\gamma, \mathsf{k\_}\gamma\}$;*
- *a **permissive set of background defeasible rule schemata** iff $B = \{\mathsf{d\_}\gamma, \mathsf{p\_}\gamma\}$ or $B = \{\mathsf{d\_}\gamma, \mathsf{k\_}\gamma\}$.*

A permissive by default set or a Kelsenian permissive set indicate that anything is defeasibly permitted. They are distinct in that a Kelsenian permissive set may better reflect the principle of prohibition as exposed by Kelsen (thus its name)[2]. Moreover, this set may appear weaker than the permissive by default set because the rule $\mathsf{k\_}\gamma$ features a point of attack $\sim \mathsf{O}\overline{\gamma}$ which rule $\mathsf{p\_}\gamma$ does not present. We will use in this paper this permissive set of background defeasible rules for our illustrations. For both sets, although one may presuppose that their rules can be used to fill normative gaps, we will see that such background rules are actually not enough to obtain normative completeness when using bivalent statement labelling semantics. Our solution to this issue will turn out to yield the same results for both sets when determining acceptance statuses of statements.

Whatever the set of background defeasible rules, we will ground the rules over a set of propositions. For our purposes, we do so over propositions of an input (domain-dependent) defeasible theory.

**Definition 4.4.** *A set of rules is a **set of background rules** with respect to a defeasible theory $T$ and a set of background defeasible rule schemata $B$, denoted* $\mathrm{BackRules}(T,B)$*, iff*

- $\mathrm{BackRules}(T,B) = \{\mathsf{d\_}\gamma, \mathsf{d\_}\overline{\gamma}, \mathsf{p\_}\gamma, \mathsf{p\_}\overline{\gamma} \mid \gamma \in \mathrm{Prop}(T)\}$ *if $B$ is a permissive by default set of background defeasible rule schemata;*
- $\mathrm{BackRules}(T,B) = \{\mathsf{d\_}\gamma, \mathsf{d\_}\overline{\gamma}, \mathsf{k\_}\gamma, \mathsf{k\_}\overline{\gamma} \mid \gamma \in \mathrm{Prop}(T)\}$ *if $B$ is a Kelsenian permissive set of background defeasible rule schemata.*

For the sake of simplicity, norms potentially captured by 'modalised rules' (e.g, rules of the form $\mathsf{O}(r : \varphi_1, \ldots, \varphi_n, \sim \varphi'_1, \ldots, \sim \varphi'_m \Rightarrow \varphi)$) are not accounted for in this paper. Such constructs and their meanings are left to possible developments of the system, towards for example secondary norms as 'meta-rules'.

Background rules are at the domain-independent core of deontic reasoning, and they are employed to background sets of rules.

**Definition 4.5.** *A set of rules Rules is a **backgrounded set of rules** with respect to a defeasible theory $T$ and a set of background defeasible rule schemata $B$ iff Rules $=$* $\mathrm{Rules}(T) \cup \mathrm{BackRules}(T,B)$.

[2] 'As a sanction-prescribing social order, the law regulates human behavior in two ways: in a positive sense, commanding such behavior and thereby prohibiting the opposite behavior; and, negatively, by not attaching a coercive act to a certain behavior, therefore not prohibiting this behavior and not commanding the opposite behavior. Behavior that legally is not prohibited is legally permitted in this negative sense.' [29].

**Example 1.** *Let us adapt H.L.A Hart's hypothetical [25] for our purposes: assume a policy stating that it is forbidden to enter in a park with a vehicle, unless there is an emergency. This policy may be formalised by the (foreground) defeasible theory* $\langle\{\mathsf{r}\},\emptyset,\emptyset\rangle$ *where*

$\mathsf{r}:\quad \mathsf{vehi}, \sim \mathsf{emer} \Rightarrow \mathsf{Fenter}$

*The Kelsenian permissive set of background rules with respect to the theory* $\langle\{\mathsf{r}\},\emptyset,\emptyset\}\rangle$ *includes all the following rules.*

| | | | | | |
|---|---|---|---|---|---|
| d_vehi : | Ovehi | ⇒ Pvehi | d_¬vehi : | O¬vehi | ⇒ P¬vehi |
| k_vehi : | ∼ Fvehi | ⇒ Pvehi | k_¬vehi : | ∼ F¬vehi | ⇒ P¬vehi |
| d_emer : | Oemer | ⇒ Pemer | d_¬emer : | O¬emer | ⇒ P¬emer |
| k_emer : | ∼ Femer | ⇒ Pemer | k_¬emer : | ∼ F¬emer | ⇒ P¬emer |
| d_enter : | Oenter | ⇒ Penter | d_¬enter : | O¬enter | ⇒ P¬enter |
| k_enter : | ∼ Fenter | ⇒ Penter | k_¬enter : | ∼ F¬enter | ⇒ P¬enter |

□

As a set of background rules is defined with respect to a defeasible theory $T$ and a set of background defeasible rule schemata $B$, the cardinality of the set of background rules is $2 \cdot |B| \cdot |\mathrm{Prop}(T)|$ (as illustrated in Example 1). Yet, for practical matters and especially implementation matters, we may employ only background defeasible rule schemata which are instantiated where necessary for the considered computation.

Concerning conflicts, we distinguish foreground conflicts and background conflicts. Foreground conflicts can be any kind of conflicts of the form $(\gamma,\overline{\gamma})$ or $(\mathsf{O}\gamma,\mathsf{O}\overline{\gamma})$ or $(\neg\mathsf{O}\gamma,\mathsf{O}\gamma)$ or $(\mathsf{O}\gamma,\neg\mathsf{O}\gamma)$; deontic conflicts can be visualised in the deontic square drawn in Figure 1.

**Definition 4.6.** *A conflict is a* ***foreground conflict*** *iff it is of the form* $(\gamma,\overline{\gamma})$ *or* $(\mathsf{O}\gamma,\mathsf{O}\overline{\gamma})$ *or* $(\neg\mathsf{O}\gamma,\mathsf{O}\gamma)$ *or* $(\mathsf{O}\gamma,\neg\mathsf{O}\gamma)$.

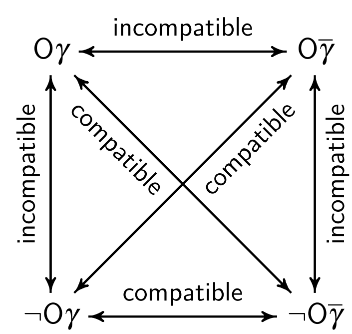


Figure 1: Deontic square of compatibility relation.

Foreground conflicts allow a knowledge engineer to specify particular conflicts between literal statements. However, specified foreground conflicts may appear incomplete in that inevitable conflicts may not be included in the foreground set. To ensure completeness of conflicts, we assume thus background conflicts.

**Definition 4.7.** *A set of conflicts is a* ***set of background conflicts*** *with respect to a defeasible theory $T$, denoted* $\mathrm{BackConflicts}(T)$, *iff* $\mathrm{BackConflicts}(T) = \{(\gamma,\overline{\gamma}),(\overline{\gamma},\gamma),(\mathsf{O}\gamma,\mathsf{O}\overline{\gamma}),(\mathsf{O}\overline{\gamma},\mathsf{O}\gamma),(\mathsf{O}\gamma,\neg\mathsf{O}\gamma),(\neg\mathsf{O}\gamma,\mathsf{O}\gamma),(\mathsf{O}\overline{\gamma},\neg\mathsf{O}\overline{\gamma}),(\neg\mathsf{O}\overline{\gamma},\mathsf{O}\overline{\gamma}) \mid \gamma \in \mathrm{Prop}(T)\}$.

Background conflicts are, so to say, domain-independent in deontic reasoning,[3] and a conflict relation is backgrounded by such conflicts if, and only if, they are included in the relation.

**Definition 4.8.** *A conflict relation $Conflicts$ is a* ***backgrounded conflict relation*** *with respect to a defeasible theory $T$ iff $Conflicts = Conflicts(T) \cup \mathrm{BackConflicts}(T)$.*

**Example 1** (continued). *The background conflict pairs are as follows.*

| | | |
|---|---|---|
| $(\mathsf{vehi},\neg\mathsf{vehi})$ | $(\mathsf{emer},\neg\mathsf{emer})$ | $(\mathsf{enter},\neg\mathsf{enter})$ |
| $(\neg\mathsf{vehi},\mathsf{vehi})$ | $(\neg\mathsf{emer},\mathsf{emer})$ | $(\neg\mathsf{enter},\mathsf{enter})$ |
| $(\mathsf{Ovehi},\mathsf{O}\neg\mathsf{vehi})$ | $(\mathsf{Oemer},\mathsf{O}\neg\mathsf{emer})$ | $(\mathsf{Oenter},\mathsf{O}\neg\mathsf{enter})$ |
| $(\mathsf{O}\neg\mathsf{vehi},\mathsf{Ovehi})$ | $(\mathsf{O}\neg\mathsf{emer},\mathsf{Oemer})$ | $(\mathsf{O}\neg\mathsf{enter},\mathsf{Oenter})$ |
| $(\mathsf{Ovehi},\neg\mathsf{Ovehi})$ | $(\mathsf{Oemer},\neg\mathsf{Oemer})$ | $(\mathsf{Oenter},\neg\mathsf{Oenter})$ |
| $(\neg\mathsf{Ovehi},\mathsf{Ovehi})$ | $(\neg\mathsf{Oemer},\mathsf{Oemer})$ | $(\neg\mathsf{Oenter},\mathsf{Oenter})$ |
| $(\mathsf{O}\neg\mathsf{vehi},\neg\mathsf{O}\neg\mathsf{vehi})$ | $(\mathsf{O}\neg\mathsf{emer},\neg\mathsf{O}\neg\mathsf{emer})$ | $(\mathsf{O}\neg\mathsf{enter},\neg\mathsf{O}\neg\mathsf{enter})$ |
| $(\neg\mathsf{O}\neg\mathsf{vehi},\mathsf{O}\neg\mathsf{vehi})$ | $(\neg\mathsf{O}\neg\mathsf{emer},\mathsf{O}\neg\mathsf{emer})$ | $(\neg\mathsf{O}\neg\mathsf{enter},\mathsf{O}\neg\mathsf{enter})$ |

□

We can remark that given any defeasible theory $T$ where conflicts are foreground or background conflicts, we have that $Conflicts(T) \subseteq \mathrm{BackConflicts}(T)$. Consequently, a conflict relation $Conflicts$ is a backgrounded conflict relation with respect to a defeasible theory $T$ if, and only if, $Conflicts = \mathrm{BackConflicts}(T)$. Thus, if one works with backgrounded conflicts of a defeasible theory, as we will do, then foreground conflicts may appear unnecessary. The definition of foreground conflicts is nevertheless formally necessary to constrain conflicts which can be given when specifying any foreground theories (as defined soon).

[3] Addendum: we can note that the definition is about sets of background conflicts and not background conflicts themselves. Nevertheless, assuming that a background conflict is defined as an element of such sets, a background conflict may be correctly qualified as a foreground conflict by Definition 4.6 (from an original position behind the deontic background for example). Such a consideration does not apply to background rules in the paper, and a more general formalisation of foreground/background elements is left to future work.

Similarly as background rules, a background conflict relation is defined with respect to a defeasible theory $T$, and the cardinality of the relation is $8 \cdot |\text{Prop}(T)|$ (as illustrated in Example 1). However, and again, we may only need background conflict schemata relationships instantiated where required.

Foreground and background deontic rules may have conflicting heads, and to ensure correct reasoning patterns, background superiorities can be proposed. At first sight, background rule $\mathsf{d\_}\gamma$ could be viewed as a very strong rule, that is, in our context a defeasible rule which is superior to any other rule and such that there exist no superior foreground rules. However, such a superiority setting does not fit well with the adopted last-link preference over arguments, as some anomalies may appear. For example, if $\mathsf{d\_}\gamma$ is superior to any foreground rules, then any argument $C$ whose top rule is $\mathsf{d\_}\gamma$ could defeat any arguments defeating the direct subargument of $C$.

**Example 2.** *For instance, suppose the following arguments:*

| | | | | | |
|---|---|---|---|---|---|
| $\mathsf{O1}:$ | | $\Rightarrow_{\mathsf{r}} \mathsf{Oa}$ | $\mathsf{O2}:$ | | $\Rightarrow_{\mathsf{r'}} \mathsf{O\neg a}$ |
| $\mathsf{P1}:$ | $\mathsf{O1}$ | $\Rightarrow_{\mathsf{d\_a}} \mathsf{Pa}$ | $\mathsf{P2}:$ | $\mathsf{O2}$ | $\Rightarrow_{\mathsf{d\_\neg a}} \mathsf{P\neg a}$ |
| $\mathsf{W1}:$ | $\sim \mathsf{Fa}$ | $\Rightarrow_{\mathsf{k\_a}} \mathsf{Pa}$ | $\mathsf{W2}:$ | $\sim \mathsf{F\neg a}$ | $\Rightarrow_{\mathsf{k\_\neg a}} \mathsf{P\neg a}$ |

*Let us assume that rule* $\mathsf{r'}$ *is superior to* $\mathsf{r}$, *i.e.* $\mathsf{r'} \succ \mathsf{r}$, *so that argument* O2 *attacks* O1. *Consequently, argument* O2 *rebuts* P1 *(on* O1*). Moreover, if rule* $\mathsf{d\_}\gamma$ *has superior or equal strength to any foreground rules, then argument* P1 *rebuts* O2 *and* P2 *(on* O2*), and* P2 *rebuts* O1 *and* P1 *(on* O1*). As a result, all arguments would be labelled* UND *as illustrated in Figure 2 on the left, but this labelling is not satisfactory. Instead, if* $\mathsf{d\_}\gamma$ *has inferior strength to any foreground rules, then we have the argumentation graph as illustrated in Figure 2 on the right, whose grounded* {IN, OUT, UND}*-labelling is more satisfactory.*

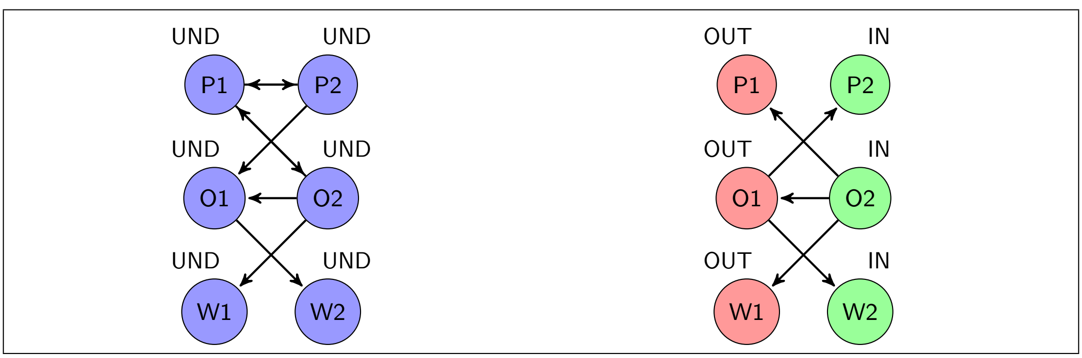


Figure 2: Grounded {IN, OUT, UND}-labellings.

□

Example 2 shows that background rule d_$\gamma$ should not be viewed as a strong rule. On the contrary, it should be conceived as a very weak rule, so that arguments with such a top rule attack no foreground arguments. In general, we assume in this paper that background rules are inferior to any foreground rules.

**Definition 4.9.** *A superiority relation is a* ***background superiority relation*** *with respect to a defeasible theory T and a set of background defeasible rule schemata B, denoted* $\mathrm{BackSup}(T,B)$*, iff* $\mathrm{BackSup}(T,B) = \{(s,r) \mid s \in \mathrm{Rules}(T), r \in \mathrm{BackRules}(T,B)\}$.

**Definition 4.10.** *A superiority relation* $\succ$ *is a* ***backgrounded superiority relation*** *with respect to a defeasible theory T and a set of background defeasible rule schemata B iff* $\succ = \succ(T) \cup \mathrm{BackSup}(T,B)$.

The definition of a backgrounded superiority relation might be simplified if strict rules were employed. However, the use of strict rules can be quite elusive [11,17], and would requisite a larger inference apparatus which would not be congruent with our design consideration on parsimony.

**Example 1** (continued). *The background pairs in the superiority relation are as follows.*

| | | |
|---|---|---|
| (r, d_vehi) | (r, d_emer) | (r, d_enter) |
| (r, k_vehi) | (r, k_emer) | (r, k_enter) |
| (r, d_¬vehi) | (r, d_¬emer) | (r, d_¬enter) |
| (r, k_¬vehi) | (r, k_¬emer) | (r, k_¬enter) |

□

A background superiority relation is defined with respect to a defeasible theory $T$ and a set of background defeasible rule schemata $B$, thus the cardinality of the relation is greater than $|B| \cdot |\mathrm{Prop}(T)| \cdot |\mathrm{Rules}(T)|$. However, similarly as background rules and background conflicts, we may only need background superiority schemata which are instantiated where necessary.

## 4.2 Deontic defeasible theory and argumentation graphs

We now can propose to 'background' defeasible theories where rules, conflicts and superiority relationships are backgrounded with respect to any foreground theory. A foreground defeasible theory is here a defeasible theory where rules are not background rules, i.e. rules whose identifiers are not identifiers of any background rules.

**Definition 4.11.** *A defeasible theory* $\langle Rules, Conflicts, \succ\rangle$ *is a* ***foreground defeasible theory*** *iff*

- *every defeasible rule in Rules is a (foreground) normative defeasible rule which is not a background defeasible rule, and*

- *every conflict in Conflicts is a foreground conflict.*

**Definition 4.12.** *A defeasible theory $\langle Rules, Conflicts, \succ \rangle$ is a* ***backgrounded defeasible theory*** *of a foreground defeasible theory T with a set of background defeasible rule schemata B iff*
- *Rules is a backgrounded set of rules with respect to T and B, and*
- *Conflicts is a backgrounded conflict relation with respect to T, and*
- *$\succ$ is a backgrounded superiority relation with respect to T and B.*

In practice, we will first write a foreground defeasible theory to then hold a backgrounded defeasible theory. In the remainder, we assume that any defeasible theory is backgrounded with a permissive set of background defeasible rule schemata, to obtain a permissive defeasible theory.

**Definition 4.13.** *A defeasible theory is a* ***permissive defeasible theory*** *iff it is a backgrounded defeasible theory with a permissive set of background defeasible rule schemata.*

From a backgrounded defeasible theory, we can build arguments. When building arguments, chaining rules implicitly uses the detachment of the consequent of rules. In that regard, we can note that deontic studies usually distinguish factual detachments and deontic detachments. For our purposes, we consider factual detachments only, leaving (defeasible) deontic detachments (if accepted) to future developments.

Once arguments are built, we can form an argumentation graph, and then label arguments and (deontic) statements to determine their statuses, as discussed next.

## 4.3 Deontic labelling semantics

On the basis of an argumentation graph built from a backgrounded defeasible theory, we can now look at semantics for (deontic) literal statements. By semantics, we mean labelling semantics as put forward in the previous section so that acceptance labellings of (deontic) literal statements are defined with respect to acceptance labellings of arguments in terms of 'if, and only if'.

First, we have to note that, sometimes, imperatives are deemed to bear no truth values, and thus no semantics in terms of truth values can be devised [27]. For instance, there is no truth value in an imperative such as 'Do not enter!'. Alternatively, however, we may evaluate the acceptance of normative statements with respect to the normative system. For example, we can evaluate whether an obligation holds in particular situation. We adopt such an epistemic view in the rest of the paper by labelling arguments and statements.

To label deontic arguments and statements, we first resort to labelling semantics as previously exposed. Hence, given the argumentation graph built from a backgrounded defeasible theory, arguments are labelled according to the grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling semantics.

Then, literal statements are labelled with statement acceptance labelling semantics. Such labellings are thus a straightforward application of standard labelling semantics corresponding to some common sense reasoning.

As such, the labelling framework satisfies some intuitive properties pertaining to deontic consistency. Let us first observe that if two arguments $A$ and $B$ have conflicting conclusions and $A$ is labelled $\mathsf{IN}$ then argument $B$ is labelled $\mathsf{OUT}$. Thus, for instance, if an $\mathsf{IN}$-labelled argument has an obligation $\mathsf{O}\gamma$ or a permission $\mathsf{P}\gamma$ for conclusion, then any argument whose conclusion is a prohibition $\mathsf{F}\gamma$ is labelled $\mathsf{OUT}$.

**Lemma 4.1.** *Let $L$ be a grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of an argumentation graph $G$ constructed from a permissive defeasible theory $\langle Rules, Conflicts, \succ\rangle$, and $A, B$ any arguments in $\mathscr{A}_G$ such that $Conflicts(\mathrm{conc}(A), \mathrm{conc}(B))$. If $L(A) = \mathsf{IN}$ then $L(B) = \mathsf{OUT}$.*

It follows that the bivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling and trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling semantics trivially imply that two conflicting deontic statements cannot be labelled $\mathsf{in}$: if a deontic statement is labelled $\mathsf{in}$ then any conflicting statement is labelled $\mathsf{ni}$ or $\mathsf{niund}$ depending on the selected statement labelling semantics.

**Proposition 4.1.** *Let $L$ be a grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of an argumentation graph constructed from a permissive defeasible theory $T$, and $\mathscr{S}$ a set of literal statements such that $\mathscr{S} = \{p, \neg p \mid p \in \mathrm{Prop}(T)\}$.*

- *Let $K$ be a bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labelling of $\mathscr{S}$ and from $\{L\}$. For any $\gamma \in \mathscr{S}$:*
  - *if $K(L, \mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \mathsf{O}\overline{\gamma}) = \mathsf{ni}$;*
  - *if $K(L, \neg\mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \mathsf{O}\gamma) = \mathsf{ni}$;*
  - *if $K(L, \mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \neg\mathsf{O}\gamma) = \mathsf{ni}$;*
- *Let $K$ be a trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling of $\mathscr{S}$ and from $\{L\}$. For any $\gamma \in \mathscr{S}$:*
  - *if $K(L, \mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \mathsf{O}\overline{\gamma}) = \mathsf{niund}$;*
  - *if $K(L, \neg\mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \mathsf{O}\gamma) = \mathsf{niund}$;*
  - *if $K(L, \mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \neg\mathsf{O}\gamma) = \mathsf{niund}$.*

*Proof.* Let us provide the proof for the first item only in the case of bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labellings (proofs for the other items follow the same structure). If $K(L, \mathsf{O}\gamma) = \mathsf{in}$ then there exists an argument $A \in \mathscr{A}_G$ such that $\mathrm{conc}(A) = \mathsf{O}\gamma$ and $L(A) = \mathsf{IN}$. Two cases: $\mathsf{O}\overline{\gamma}$ is the conclusion of an argument, or not. In the first case, by Lemma 4.1, for any argument $B \in \mathscr{A}_G$ such that $\mathrm{conc}(B) = \mathsf{O}\overline{\gamma}$, if $L(A) = \mathsf{IN}$ then $L(B) = \mathsf{OUT}$, and thus $K(L, \mathsf{O}\overline{\gamma}) = \mathsf{ni}$. In the second case, if $\mathsf{O}\overline{\gamma}$ is not the conclusion of any argument in $\mathscr{A}_G$, then $K(L, \mathsf{O}\overline{\gamma}) = \mathsf{ni}$. Therefore, if $K(L, \mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \mathsf{O}\overline{\gamma}) = \mathsf{ni}$. □

We can also remark that if an obligation $\mathsf{O}\gamma$ is labelled $\mathsf{in}$ then the implied permission $\neg\mathsf{O}\overline{\gamma}$ (i.e. $\mathsf{P}\gamma$) is also labelled $\mathsf{in}$.

**Proposition 4.2.** *Let $L$ be a grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of an argumentation graph constructed from a permissive defeasible theory $T$, $\mathscr{S}$ a set of literal statements such that $\mathscr{S} = \{p, \neg p \mid p \in \mathrm{Prop}(T)\}$, and $K$ a bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labelling or trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling of $\mathscr{S}$ and from $\{L\}$. For any $\gamma \in \mathscr{S}$: if $K(L, \mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \neg\mathsf{O}\overline{\gamma}) = \mathsf{in}$.*

*Proof.* Given an argumentation graph $G$ constructed from a permissive defeasible theory, if $K(L, \mathsf{O}\gamma) = \mathsf{in}$, then there exist an argument $A \in \mathscr{A}_G$ and $B \in \mathscr{A}_G$ such that $\mathrm{conc}(A) = \mathsf{O}\varphi$ and $B : A \Rightarrow_{\mathsf{d\_}\gamma} \mathsf{P}\gamma$. Let $\mathscr{A}^{\rightsquigarrow} \subseteq \mathscr{A}_G$ be the set of attackers of $A$, $\mathscr{A}^{\leftsquigarrow} \subseteq \mathscr{A}_G$ the set of arguments attacked by $A$, and $\mathscr{B}^{\rightsquigarrow} \subseteq \mathscr{A}_G$ the set of attackers of $B$. We have that $\mathscr{B}^{\rightsquigarrow} \subseteq \mathscr{A}^{\rightsquigarrow} \cup \mathscr{A}^{\leftsquigarrow}$. By Definition 3.12, if $L(A) = \mathsf{IN}$ then for any argument $C \in \mathscr{A}^{\rightsquigarrow} \cup \mathscr{A}^{\leftsquigarrow}$ $L(C) = \mathsf{OUT}$. Since $\mathscr{B}^{\rightsquigarrow} \subseteq \mathscr{A}^{\rightsquigarrow} \cup \mathscr{A}^{\leftsquigarrow}$, for any argument $C \in \mathscr{B}^{\rightsquigarrow}$ $L(C) = \mathsf{OUT}$, and thus $L(B) = \mathsf{IN}$. Therefore, if $L(A) = \mathsf{IN}$ then $L(B) = \mathsf{IN}$, and thus if $K(L, \mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \mathsf{P}\gamma) = \mathsf{in}$, i.e. if $K(L, \mathsf{O}\gamma) = \mathsf{in}$ then $K(L, \neg\mathsf{O}\overline{\gamma}) = \mathsf{in}$. □

Legal scholars, however, may argue that bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labellings are not satisfactory. Such labellings are not legally satisfactory, because, given an argumentation graph from any backgrounded defeasible theory, and though we have arguments supporting weak permissions thanks to background rules, it may the case that all arguments are labelled $\mathsf{UND}$ and consequently, using a bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labelling, deontic statements $\mathsf{O}\gamma$, $\mathsf{F}\gamma$ and $\mathsf{P}\gamma$ may be labelled $\mathsf{ni}$, see Example 3. Such labelling outcomes of common sense appear thus inappropriate to address normative completeness. For this reason, we discard bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labelling semantics and put forward trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling to cater for deontic reasoning and in particular normative completeness.

**Example 3.** *Let us consider the following arguments, along with the associated argumentation graph and grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling drawn in Figure 3.*

| | | | | | |
|---|---|---|---|---|---|
| $\mathsf{O1}:$ | | $\Rightarrow_{\mathsf{r}} \mathsf{Oa}$ | $\mathsf{O2}:$ | | $\Rightarrow_{\mathsf{r'}} \mathsf{O}\neg\mathsf{a}$ |
| $\mathsf{P1}:$ | $\mathsf{O1}$ | $\Rightarrow_{\mathsf{d\_a}} \mathsf{Pa}$ | $\mathsf{P2}:$ | $\mathsf{O2}$ | $\Rightarrow_{\mathsf{d\_\neg a}} \mathsf{P}\neg\mathsf{a}$ |
| $\mathsf{W1}:$ | $\sim \mathsf{Fa}$ | $\Rightarrow_{\mathsf{k\_a}} \mathsf{Pa}$ | $\mathsf{W2}:$ | $\sim \mathsf{F}\neg\mathsf{a}$ | $\Rightarrow_{\mathsf{k\_\neg a}} \mathsf{P}\neg\mathsf{a}$ |

*First, a naive common sense bivalent reasoning can be captured by the acceptance bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labelling $\langle \emptyset, \{\mathsf{Oa}, \mathsf{Pa}, \mathsf{O}\neg\mathsf{a}, \neg\mathsf{Pa}\}\rangle$. However, this bivalent labelling is problematic from a legal stance because statement $\mathsf{a}$ is here neither obligated, nor permitted nor prohibited. To address this gap, we can employ a trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling $\langle \emptyset, \{\mathsf{Oa}, \mathsf{Pa}, \mathsf{O}\neg\mathsf{a}, \neg\mathsf{Pa}\}, \emptyset\rangle$ according to which the deontic status of $\mathsf{a}$ is undecided.* □

More formally, the definition of normative gaps, as we may conceive it in terms of statement labellings, depends on whether bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labellings or trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labellings are employed.

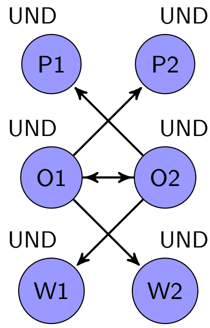


Figure 3: Grounded {IN, OUT, UND}-labelling.

**Definition 4.14.** *Let $L$ be a grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of an argumentation graph constructed from a permissive defeasible theory $T$, and the set of literal statements $\mathscr{S} = \{p, \neg p \mid p \in \text{Prop}(T)\}$.*

- *Let $K$ denote a bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labelling of $\mathscr{S}$ and from $\{L\}$. There is a $\{\mathsf{in}, \mathsf{ni}\}$**-labelling normative gap** iff there exists a literal $\gamma \in \mathscr{S}$ such that*

$$K(L, \mathsf{O}\gamma) = \mathsf{ni} \text{ and } K(L, \mathsf{O}\overline{\gamma}) = \mathsf{ni} \text{ and } K(L, \neg\mathsf{O}\overline{\gamma}) = \mathsf{ni}.$$

- *Let $K'$ denote a trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling of $\mathscr{S}$ and from $\{L\}$. There is a $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$**-labelling normative gap** iff there exists a literal $\gamma \in \mathscr{S}$ such that*

$$K'(L, \mathsf{O}\gamma) = \mathsf{niund} \text{ and } K'(L, \mathsf{O}\overline{\gamma}) = \mathsf{niund} \text{ and } K'(L, \neg\mathsf{O}\overline{\gamma}) = \mathsf{niund}.$$

As illustrated in Example 3, bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labellings may lead to $\{\mathsf{in}, \mathsf{ni}\}$-labelling normative gaps, whereas trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labellings can address normative completeness. To understand why, we can first observe that any backgrounded defeasible theory along with a trivalent labelling semantics lead to a third interpretation of the principle of prohibition in terms of labelling: if something is not prohibited (the prohibition is labelled niund) then it is permitted (the permission is labelled in).

**Proposition 4.3.** *Let $L$ be a grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of an argumentation graph constructed from a permissive defeasible theory $T$, $\mathscr{S}$ a set of literal statements such that $\mathscr{S} = \{p, \neg p \mid p \in \text{Prop}(T)\}$, and $K$ a trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling of $\mathscr{S}$ and from $\{L\}$. For any $\gamma \in \mathscr{S}$: if $K(L, \mathsf{F}\gamma) = \mathsf{niund}$ then $K(L, \mathsf{P}\gamma) = \mathsf{in}$.*

*Proof.* Given an argumentation graph constructed from a permissive defeasible theory $T$, for any $\gamma \in \mathscr{S} = \{p, \neg p \mid p \in \text{Prop}(T)\}$, there exists a unique argument $W : \sim \mathsf{F}\gamma \Rightarrow_{\mathsf{k\_}\gamma} \mathsf{P}\gamma$ or ($W : \quad \Rightarrow_{\mathsf{p\_}\gamma} \mathsf{P}\gamma$). All attackers of $W$ are arguments whose conclusion is $\mathsf{F}\gamma$ (i.e. $\mathsf{O}\overline{\gamma}$). If $K(L, \mathsf{O}\overline{\gamma}) = \mathsf{niund}$ then all attackers of $W$ are OUT, and thus $W$ is labelled IN, and $\mathsf{P}\gamma$ is labelled in. i.e. $K(L, \neg\mathsf{O}\overline{\gamma}) = \mathsf{in}$. Therefore, if $K(L, \mathsf{F}\gamma) = \mathsf{niund}$ then $K(L, \mathsf{P}\gamma) = \mathsf{in}$. □

On the basis of this interpretation in terms of labelling of the principle of prohibition, we now can easily show that trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labellings address normative completeness.

**Theorem 4.1.** *Let $L$ be a grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of an argumentation graph constructed from a permissive defeasible theory $T$, $\mathscr{S}$ a set of literal statements such that $\mathscr{S} = \{p, \neg p \mid p \in \mathrm{Prop}(T)\}$, and $K$ a trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling of $\mathscr{S}$ and from $\{L\}$. For any $\gamma \in \mathscr{S}$:*

$$K(L, \mathsf{O}\gamma) \neq \mathsf{niund} \textit{ or } K(L, \mathsf{O}\overline{\gamma}) \neq \mathsf{niund} \textit{ or } K(L, \neg\mathsf{O}\overline{\gamma}) \neq \mathsf{niund}.$$

*Proof.* There are three cases: 1. $K(L, \mathsf{O}\overline{\gamma}) = \mathsf{in}$, 2. $K(L, \mathsf{O}\overline{\gamma}) = \mathsf{und}$, and 3. $K(L, \mathsf{O}\overline{\gamma}) = \mathsf{niund}$. In this last case, by Proposition 4.3, $K(L, \mathsf{P}\gamma) = \mathsf{in}$, i.e. $K(L, \neg\mathsf{O}\overline{\gamma}) = \mathsf{in}$. Therefore in any case, $K(L, \mathsf{O}\gamma) \neq \mathsf{niund}$ or $K(L, \mathsf{O}\overline{\gamma}) \neq \mathsf{niund}$ or $K(L, \neg\mathsf{O}\overline{\gamma}) \neq \mathsf{niund}$. □

Hence, trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labellings address normative completeness by means of the status ‘undecided’ for deontic statements. Eventually, we can remark that such cases of undecidedness can be disentangled in various ways, typically by a competent authority, e.g. a judge.

Above results hold for any permissive theory. Consequently, they hold for backgrounded defeasible theory of any (foreground) defeasible theory with a permissive by default set of background defeasible rule schemata, or with a Kelsenian permissive set of background defeasible rule schemata. In general, it turns out that both sets yield the same trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling.

**Theorem 4.2.** *Let*
- *$T$ be a (foreground) defeasible theory;*
- *$U$ be the backgrounded defeasible theory of $T$ with a permissive by default set of background defeasible rule schemata;*
- *$V$ be the backgrounded defeasible theory of $T$ with a Kelsenian permissive set of background defeasible rule schemata;*
- *$L_U$ the grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of the argumentation graph constructed from $U$;*
- *$L_V$ the grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of the argumentation graph constructed from $V$;*
- *$\mathscr{S}$ a set of literal statements such that $\mathscr{S} = \{p, \neg p \mid p \in \mathrm{Prop}(T)\}$, and $K$ a trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling of $\mathscr{S}$ and from $\{L\}$.*

*For any $\gamma \in \mathscr{S}$:*

$$K(L_U, \gamma) = K(L_V, \gamma) \textit{ and } K(L_U, \mathsf{O}\gamma) = K(L_V, \mathsf{O}\gamma) \textit{ and } K(L_U, \neg\mathsf{O}\gamma) = K(L_V, \neg\mathsf{O}\gamma).$$

*Proof.* Let $GU$ ($GV$ resp.) denote the argumentation graph constructed from $U$ ($V$ resp.). Let $\mathsf{f}$ be the bijection such that $\mathsf{f} : \mathscr{A}_{GU} \to \mathscr{A}_{GV}$ and $\mathsf{f}(A) = B$ iff

- if $A$ is of the form $A: \ \Rightarrow_{\mathsf{p\_}\gamma} \mathsf{P}\gamma$ then $B$ is of the form $B: \ \sim \mathsf{F}\gamma \Rightarrow_{\mathsf{k\_}\gamma} \mathsf{P}\gamma$, and
- if $A$ is of the form $A: A_1, \dots A_n, \sim \varphi_1, \dots, \sim \varphi_m \Rightarrow_r \varphi$ ($r \neq \mathsf{p\_}\gamma$) then $B$ is of the form $B: \mathsf{f}(A_1), \dots \mathsf{f}(A_n), \sim \varphi_1, \dots, \sim \varphi_m \Rightarrow_r \varphi$.

In addition, $(A, A') \in \rightsquigarrow_{GU}$ iff $(\mathsf{f}(A), \mathsf{f}(A')) \in \rightsquigarrow_{GV}$, and thus ($GU$ and $GV$ are isomorphic and) $L_U(A) = L_V(\mathsf{f}(A))$. Then, as $\mathrm{conc}(A) = \mathrm{conc}(\mathsf{f}(A))$, for any $\gamma \in \mathscr{S}$: $K(L_U, \gamma) = K(L_V, \gamma)$, $K(L_U, \mathsf{O}\gamma) = K(L_V, \mathsf{O}\gamma)$, and $K(L_U, \neg\mathsf{O}\gamma) = K(L_V, \neg\mathsf{O}\gamma)$. □

Hence, the adoption of the principle of prohibition 'anything that is not prohibited is permitted' as a schema $\mathsf{k\_}\gamma : \sim \mathsf{O}\overline{\gamma} \Rightarrow \mathsf{P}\gamma$ or the use of a statement such as 'anything is permitted prima facie' as a schema $\mathsf{p\_}\gamma: \ \Rightarrow \mathsf{P}\gamma$ are two alternatives to cater for normative completeness, and both alternatives actually lead to the same results in terms of statement labellings.

Finally, concerning weak and strong permissions, the reification of doctrinal pieces into defeasible theories blurs somewhat the distinction based on conventional definitions. For example, a permission $\mathsf{P}\gamma$ which is the conclusion of an IN-labelled argument $W: \ \Rightarrow_{\mathsf{p\_}\gamma} \mathsf{P}\gamma$ is necessarily labelled in. Consequently we may say that such a permission is derived from a rule and thus, it is a strong permission by definition, whereas it is a weak permission from its conception. For this reason, a third kind of permission may be introduced, which we may call 'doctrinal permission', since such a doctrinal permission for something is derived from the non-existence or rejection of its prohibition and on the basis of reified doctrinal principles.

# 5 Illustration

To illustrate our system, let us reappraise the policy stating that it is forbidden to enter in a park with a vehicle, unless there is an emergency. This policy and the assumptions may be formalised in different ways. We illustrate our system with one option which is developed in the remainder of the section.

## 5.1 Backgrounded defeasible theory

We assume that there is a vehicle at the entrance of the park, and that there may be an emergency, maybe not (in the Hart-Fuller debate [19, 25], uncertainty was originally about what can be classified as a vehicle). Let us capture this with the foreground defeasible theory $\langle \{\mathsf{rv}, \mathsf{re}, \mathsf{r\overline{e}}, \mathsf{r}\}, \emptyset, \emptyset \rangle$ where

$$\begin{array}{ll} \mathsf{rv} & \Rightarrow \mathsf{vehi} \\ \mathsf{re} & \Rightarrow \mathsf{emer} \\ \mathsf{r\bar{e}} & \Rightarrow \neg\mathsf{emer} \\ \mathsf{r:} & \mathsf{vehi}, \sim \mathsf{emer} \Rightarrow \mathsf{Fenter} \end{array}$$

Let us adopt a Kelsenian permissive set of background defeasible rule schemata. The foreground theories can be then backgrounded to yield a backgrounded theory featuring, amongst others, background rules as exposed in Example 1.

## 5.2 Argument and argumentation graph construction

We can construct the following arguments from background rules:

$$\begin{array}{llll|llll} \mathsf{W1:} & \sim \mathsf{Fvehi} & \Rightarrow_{\mathsf{k\_vehi}} & \mathsf{Pvehi} & \mathsf{W4:} & \sim \mathsf{F\neg vehi} & \Rightarrow_{\mathsf{k\_\neg vehi}} & \mathsf{P\neg vehi} \\ \mathsf{W2:} & \sim \mathsf{Femer} & \Rightarrow_{\mathsf{k\_emer}} & \mathsf{Pemer} & \mathsf{W5:} & \sim \mathsf{F\neg emer} & \Rightarrow_{\mathsf{k\_\neg emer}} & \mathsf{P\neg emer} \\ \mathsf{W3:} & \sim \mathsf{Fenter} & \Rightarrow_{\mathsf{k\_enter}} & \mathsf{Penter} & \mathsf{W6:} & \sim \mathsf{F\neg enter} & \Rightarrow_{\mathsf{k\_\neg enter}} & \mathsf{P\neg enter} \end{array}$$

In addition, we can build the following arguments from the foreground rules and rule $\mathsf{d\_\neg enter}$:

$$\begin{array}{llll|lll} \mathsf{A1:} & & \Rightarrow_{\mathsf{rv}} & \mathsf{vehi} & \mathsf{B1} & \Rightarrow_{\mathsf{re}} & \mathsf{emer} \\ \mathsf{A2:} & \mathsf{A1}, \sim \mathsf{emer} & \Rightarrow_{\mathsf{r}} & \mathsf{Fenter} & \mathsf{C1} & \Rightarrow_{\mathsf{r\bar{e}}} & \neg\mathsf{emer} \\ \mathsf{A3:} & \mathsf{A2} & \Rightarrow_{\mathsf{d\_\neg enter}} & \mathsf{P\neg enter} & & & \end{array}$$

Consequently, we can form the argumentation graph $\mathsf{G}$ such that: $\mathscr{A}_{\mathsf{G}} = \{\mathsf{A1}, \mathsf{A2}, \mathsf{A3}, \mathsf{B1}, \mathsf{C1}, \mathsf{W1}, \mathsf{W2}, \mathsf{W3}, \mathsf{W4}, \mathsf{W5}, \mathsf{W6}\}$, and $\rightsquigarrow_{\mathsf{G}} = \{(\mathsf{B1},\mathsf{C1}), (\mathsf{C1},\mathsf{B1}), (\mathsf{B1}, \mathsf{A2}), (\mathsf{B1},\mathsf{A3}), (\mathsf{A2},\mathsf{W3})\}$, see Figure 4.

We can note that we have built arguments to support weak/doctrinal permissions, thus we can argue and present full-fledged arguments about such permissions.

## 5.3 Argument and statement labellings

Let $\mathsf{L1}$ denote the grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of argumentation graph $\mathsf{G}$ (as illustrated in Figure 4). Accordingly, we can lay the bivalent $\{\mathsf{in}, \mathsf{ni}\}$-labelling and trivalent $\{\mathsf{in}, \mathsf{und}, \mathsf{niund}\}$-labelling as in Table 1.

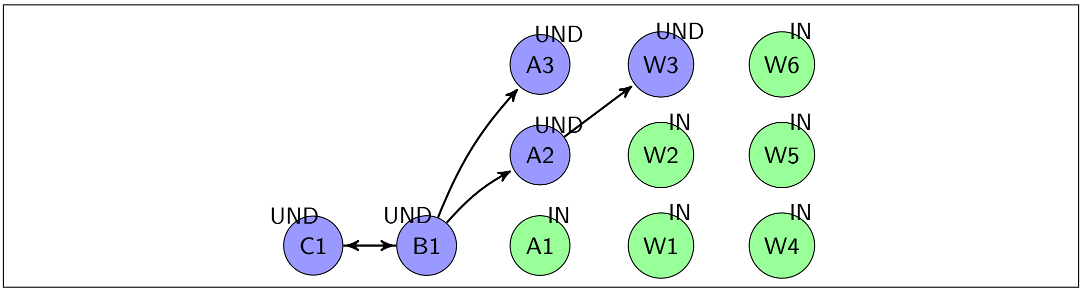


Figure 4: Grounded {IN, OUT, UND}-labelling of argumentation graph G.

| | vehi | emer | ¬emer | enter | ¬enter |
|---|---|---|---|---|---|
| $K(\mathsf{L1}, \cdot)$ | in | ni | ni | ni | ni |
| $K(\mathsf{L1}, \cdot)$ | in | und | und | niund | niund |

| | Ovehi | Oemer | O¬emer | Oenter | O¬enter |
|---|---|---|---|---|---|
| $K(\mathsf{L1}, \cdot)$ | ni | ni | ni | ni | ni |
| $K(\mathsf{L1}, \cdot)$ | niund | niund | niund | niund | und |

| | Pvehi | Pemer | P¬emer | Penter | P¬enter |
|---|---|---|---|---|---|
| $K(\mathsf{L1}, \cdot)$ | in | in | in | ni | in |
| $K(\mathsf{L1}, \cdot)$ | in | in | in | und | in |

Table 1: Bivalent {in, ni}-labelling and trivalent {in, und, niund}-labelling.

The {in, no}-bivalent labelling results into a normative gap (the statement enter is neither obligatory nor prohibited nor permitted), whereas the trivalent {in, und, niund}-labelling fills the gap by labelling the permission to enter as undecided.

### 5.4 Violation and contrary-to-duty obligation

Let us extend the illustration with the formalisation of a violation and a contrary-to-duty obligation. Contrary-to-duty obligations can be a pitfall for deontic formalisms which have a more sophisticated conception of deontic modalities [13, 35], and we would like to illustrate how such obligations can be handled in our argumentation formalism within our KB approach.

Let us suppose that the park policy also states that a violation of the prohibition would be sanctioned by a fine (the amount does not matter for our purposes). To capture such a policy, we can add the following rules.

$\mathsf{v}:\quad \mathsf{Fenter, enter} \Rightarrow \mathsf{violation} \qquad \mathsf{v}':\quad \Rightarrow \neg\mathsf{violation}$
$\mathsf{f}:\quad \mathsf{violation} \Rightarrow \mathsf{fine} \qquad \mathsf{f}':\quad \Rightarrow \neg\mathsf{fine}$

such that $\mathsf{v} \succ \mathsf{v}'$ and $\mathsf{f} \succ \mathsf{f}'$. Rules $\mathsf{v}'$ and $\mathsf{f}'$ specify that, by default, we can derive that there is neither violation nor fine, unless the contrary is shown.

Furthermore, a new park management can add a contrary-to-duty obligation: if the prohibition is violated then one should stop driving in the park. We can thus add the following rule.

$\mathsf{s}:\quad \mathsf{violation} \Rightarrow \mathsf{Ostop}$

A sequence of compensatory obligations can be added along similar lines. Eventually, we also assume that the vehicle enters in the park:

$\mathsf{e}:\quad \Rightarrow \mathsf{enter}$

Let L2 denote the grounded {IN, OUT, UND}-labelling of the implied argumentation graph. The acceptance {in, no}-bivalent labelling and trivalent {in, und, niund}-labelling of new statements are exposed in Table 2.

| | violation | ¬violation | fine | ¬fine | stop | ¬stop |
|---|---|---|---|---|---|---|
| $K(\mathsf{L2},\cdot)$ | ni | ni | ni | ni | ni | ni |
| $K(\mathsf{L2},\cdot)$ | und | und | und | und | niund | niund |
| | **Oviolation** | **O¬violation** | **Ofine** | **O¬fine** | **Ostop** | **O¬stop** |
| $K(\mathsf{L2},\cdot)$ | ni | ni | ni | ni | ni | ni |
| $K(\mathsf{L2},\cdot)$ | niund | niund | niund | niund | und | niund |
| | **Pviolation** | **P¬violation** | **Pfine** | **P¬fine** | **Pstop** | **P¬stop** |
| $K(\mathsf{L2},\cdot)$ | in | in | in | in | in | ni |
| $K(\mathsf{L2},\cdot)$ | in | in | in | in | in | und |

Table 2: Bivalent {in, ni}-labelling and trivalent {in, und, niund}-labelling.

Again, we can see that the {in, no}-bivalent labelling leads to a normative gap (¬stop is neither obligatory nor prohibited nor permitted), whereas the trivalent {in, und, niund}-labelling fills the gap by labelling the permission to not stop (i.e. the permission to drive through the park) as undecided.

## 6 Some Design Evaluation

Let us briefly discuss to what extent the framework meets the design considerations given in Section 2, that is, considerations on human interface (explainability, emulation, isomorphism), and inference (parsimony, modularity and computational efficiency), and a legit requirement regarding normative completeness.

**Explainability** An argumentation framework can inherently ease explanations of acceptance statuses of statements by presenting relevant arguments. The acceptance labelling of arguments is then exposed by the grounded labelling, possibly through a relatively simple algorithm (Algorithm 1) and resulting appealing graphical representations. A distinctive feature of the proposed system is that arguments can be built in support or against permissions and in particular weak/doctrinal permissions.
**Emulation** The system emulates the way humans argue about norms to the extent that modus ponens is endorsed as a natural reasoning step to build arguments, that attacks amongst arguments are meaningful, and that the sequential multi-labelling model featuring argument and argumentation graph production, argument acceptance/justification, statement acceptance/justification reflects a well-ordered reasoning based on arguments. Beyond that, emulation is further met in that the proposal employs the grounded semantics for which there exist simple dialogical argument games, see e.g. [12, 36, 38], in which full-fledged arguments supporting weak/doctrinal permissions can be put forward.
**Isomorphism** The system allows an isomorphic representation of normative systems as long as norms can be isomorphically represented by (foreground) defeasible rules. In that regard, we have proposed to have both the ‘unless’ conjunction ($\sim$) and a superiority relation over rules to get more flexibility to represent norms, thereby yielding a fine-grained expressiveness (as illustrated in Section 5). Furthermore, background rules can be viewed isomorphic to some (possibly very basic) deontic doctrine. The system can be critised too. For example, criminal codes usually do not hold norms with explicit expressions of conditional prohibitions: they usually directly express sanctions with regard to certain conducts, while our system would typically require the expression of prohibitions for automated reasoning.

**Parsimony** Parsimony can be evaluated at the levels of the inference machiney and knowledge representation. At the level of knowledge representation, parsimony is affected in that core deontic reasoning have been captured by background rules, conflicts and superiority relation to background theories. Deontic reasoning is also captured at the level of statement labellings to address normative gaps, without any impacts on the knowledge representation. At the inference level, the system is parsimonious in that arguments are built with defeasible rules and factual detachment only. Of course, other constructs such as strict rules, ‘defeaters’ or facts might bring a gain of expressiveness to the framework; the evaluation of such gain is left to further investigations. The system is also parsimonious in that argument acceptance and justification stages coincide, essentially because grounded $\{\mathsf{IN},\mathsf{OUT},\mathsf{UND}\}$-labelling is utilised. Importantly, parsimony was required to ease combination with other developments in argumentation. In that respect, the setting can be straightforwardly subject to probabilistic enhancements through, for example, probabilistic labellings, and thus machine learning endeavours, see e.g. [42, 43].

**Modularity** The modularity of the system is largely based on (the instantiation of) the multi-labelling model for argumentation. At a theoretical level, we can use variants in argument and statement labelling semantics, as evidenced by our use of bivalent and trivalent labellings. At a more practical level, as the multi-labelling model supports a separation of concerns corresponding to different stages of the argumentation process, we can decompose an 'argument-based software system' into well-defined independent modules, for example by developing a module for each labelling stage, thereby easing verification, validation and maintenance of such a system.

**Efficiency** Concerning computationtal complexity, the reason to focus on the grounded $\{\mathsf{IN},\mathsf{OUT},\mathsf{UND}\}$-labelling is that it is unique and it can be computed in a polynomial time, see Algorithm 1. Once the grounded $\{\mathsf{IN},\mathsf{OUT},\mathsf{UND}\}$-labelling is computed, one can trivially compute any acceptance trivalent $\{\mathsf{in},\mathsf{und},\mathsf{niund}\}$-labellings and other statement labellings. Hence given the argumentation graph built from a permissive defeasible theory, the overall time complexity to compute the labelling over a set of statements is polynomial. However, as previously mentioned, the number of arguments in an argumentation graph constructed from a defeasible theory may not be polynomial in the number of rules of the theory. If we focus on those theories from which argumentation graphs can be constructed efficiently, then argumentation graphs construction, argument labellings and statement labellings can be achieved efficiently, and thus the overall system is efficient in such cases.

**Completeness** Normative completeness has been primarily addressed by endorsing the principle of prohibition. The principle has been interpreted in three different ways. First, an interpretation has been given as a blunt syntactical equivalence between $\neg\mathsf{O}\overline{\gamma}$ and $\mathsf{P}\gamma$ (Notation 4.1). Second, the principle has been read as a background rule $\mathsf{k\_}\gamma : \sim \mathsf{O}\overline{\gamma} \Rightarrow \mathsf{P}\gamma$. Finally, the principle has been interpreted in terms of statement labellings (Proposition 4.3). As bivalent statement labellings are not sufficient to obtain normative completeness through the principle, we have supplemented it with a trivalent labelling semantics to deal with cases of undecidedness. Eventually, we have to note that the principle is shared by various legal systems, however the way completeness is resolved depends much on the considered legal system.

This brief evaluation is inherently partial as it is limited to the elicited requirements, see e.g. [18] for some formal issues. A more complete evaluation is left to future investigations, possibly in light of comparison with related implementations. An overview of related work is given next.

# 7 Related Work

There exists an increasing amount of work to capture normative reasoning through argumentation or non-monotonic frameworks akin to argumentation, see [10] for a systematic account of legal reasoning and argumentation from a logical, philosophical and legal perspective. Let us focus on some formal works related to our undertaking,

ASPIC$^+$ argumentation framework has first been exploited to express arguments about norms as the application of argument schemes to knowledge bases of facts and norms [39]. However, in [39], norms are expressed without any deontic operators for specifying obligations, permissions and prohibitions. The work was thereupon reappraised by L. van der Torre and S. Villata in [48] to integrate deontic modalities, by adopting input/output logic [34] for the analysis. Despite appearances, there are many differences with our undertaking: the most obvious are briefly exposed here. Conditional obligations and conditional permissions are represented by rules of the form $L_1 \wedge \ldots \wedge L_n \leadsto OL$ and $L_1, \ldots, L_n \leadsto \neg OL$ respectively, where $L$'s are literals. Thus, obligations and permissions do not appear in the antecedents of norms. Moreover, the system in [48] does not deal with weak permissions. Furthermore, the conjunction 'unless' ($\sim$) is not used in conditionals. Eventually, neither argument nor statement labelling semantics are specified in [48], though such semantics could be easily integrated in the system.

In another line of research, Beirlaen et al. presents in [6] a formal argumentation system for dealing with the detachment of prioritised conditional obligations and permissions. To do so, Beirlaen et al. devise an argumentation framework where arguments are proof sequences, and they employ Dung's grounded semantics to determine accepted arguments. Again, there are many differences with our work, we expose the most obvious here. A first difference concerns the language. In [6], the language pertains to a modal extension of propositional classical logic, whereas our language is restricted to literals supplemented by deontic operators. Conditional obligations and conditional permissions are denoted by formulas $A \Rightarrow_{\mathsf{O}} B$ and $A \Rightarrow_{\mathsf{P}} B$ respectively, where $A$ and $B$ are propositional formulas. Consequently, neither obligations nor permissions can appear in the antecedents of norms in [6]. Moreover, conditionals in [6] do not cater for the conjunction 'unless' as we do. Then, a major difference holds in that conditional obligations and conditional permissions in [6] are associated with a degree of priority, which is taken into account to define defeats between arguments. In our work, conditionals are directly prioritised through a superiority relation. Beirlaen et al. focus on extension-based grounded semantics, whereas we employ grounded labelling semantics to obtain more refined argument labellings which allow us to implement trivalent statement labellings. Computation complexity is not considered in [6], whereas we have restricted our framework on labellings possibly computed with efficient algorithms.

Another endeavour in defeasible normative reasoning rests on variants of Defeasible Logic (DL), see e.g. [23, 24]. The work includes rich deontic constructs such as temporal deontic modalities and sequence of compensatory obligations. However, DL is not without weaknesses. For example, DL features strict rules, but does not satisfy closure under strict rules [11]. In that regard, our work solves the issue by simply discarding strict rules. DL also features so-called ‘defeaters’, i.e. rules which cannot be used in arguments supporting a conclusion, and it can be challenging to find succinct counterparts in natural languages for such defeaters. Isomorphism can thus be questioned in DL. Compared to the work reported here, a major difference holds in that DL variants capture legal reasoning patterns in proof theories and thus at the inference level. In that regard, we can say that the loss of parsimony at the knowledge representation level in the framework reported here has been compensated by a gain of parsimony at the inference model. Concerning semantics, a substantial difference with our work holds in that there is no ‘undecided’ status to tag statements. For example, if a defeasible theory comprises two rules $r: \;\; \Rightarrow_O c$ and $s: \;\; \Rightarrow_O \neg c$ (capturing that $c$ and $\neg c$ are both obligatory) with no superiority relations and no facts, then the deontic development of DL in [22] does not tag the obligations of $c$ and $\neg c$ as undecided but tags $c$ and $\neg c$ as not obligatory ($-\partial_O c$ and $-\partial_O \neg c$), and eventually it derives $c$ and $\neg c$ as permitted ($+\partial_P c$ and $+\partial_P \neg c$). However, such tagging would be inappropriate in applications where permissions are not entailed from conflicting obligations. Eventually, argumentation semantics exist for DL [21,30] (with no status for ‘undecidedness’) but no counterparts have been developed for deontic variants.

In comparison to the above-mentioned works, substantial differences hold in that we have reified doctrinal prices to build our deontic rule-based argumentation system. We have shown that standard bivalent statement labelling semantics fall short to deal with normative completeness, and we have proposed a trivalent statement labelling semantics to address this point.

## 8 Conclusion

A deontic rule-based argumentation system has been devised to represent and reason upon conditional norms featuring obligations, prohibitions and (strong or weak) permissions. To do so, we have proposed the use of defeasible rule schemata to the greatest extent to capture deontic patterns. By doing so, we could straighforwardly adopt a common model consisting of three stages [4], namely argument and argumentation graph production, argument acceptance/justification and statement acceptance/justification. More specifically, given an argumentation graph, we have proposed to label arguments using grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling semantics, and given the grounded $\{\mathsf{IN}, \mathsf{OUT}, \mathsf{UND}\}$-labelling of the graph then we have proposed to simply label (deontic) statements using a trivalent labelling semantics.

We have learnt that it is possible to build deontic argumentation frameworks capturing conditional norms on the basis of common constructs from the literature on argumentation.

In particular, the system uses (two possible sets of) inference rule schemata, and only one single (implicit) inference rule, namely (defeasible) modus ponens. In this setting, we have learnt that a standard bivalent labelling may appear insufficient to cover aspects of normative completeness, and that this issue can be addressed by using a trivalent labelling. Through our KB approach, we have retrieved three possible interpretations of the principle of prohibition: as a syntactical equivalence, as a defeasible rule, and as a proposition on the labelling of deontic statement labellings. We have also shown that the adoption of the principle of prohibition 'anything that is not prohibited is permitted' as a schema or the use of a statement such as 'anything is permitted prima facie' as another schema are two alternatives to address normative completeness, and both alternatives actually lead to the same results in terms of statement labellings. Eventually, concerning weak and strong permissions, the reification of doctrinal pieces into defeasible theories blurs somewhat the distinction based on conventional definitions. We have thus introduced a third kind of permission, called doctrinal permission: a doctrinal permission for something is derived from the non-existence or rejection of its prohibition and on the basis of reified doctrinal principles.

Multiple developments are possible. Firstly, the principle of prohibition is not applicable in any normative system. Consequently, the framework can be further developed where the principle cannot/should not be applied. Secondly, in order to enrich the framework, other normative constructs and reasoning patterns may be integrated to build arguments, and labelling variants can be explored. Finally, the system is meant to be combined with other developments in argumentation, ranging from probabilistic argumentation to argumentation in (normative) multi-agent systems.

# References

[1] Carlos E. Alchourrón and Eugenio Bulygin. *Normative systems*. Springer-Verlag, 1971.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.

[3] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. Abstract Argumentation Frameworks and Their Semantics. In P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, editors, *Handbook of Formal Argumentation*, pages 159–236. College Publications, 2018.

[4] Pietro Baroni, Guido Governatori, and Régis Riveret. On labelling statements in multi-labelling argumentation. In *Proc. of the 22nd Euro. Conf. on Artificial Intelligence*, pages 489–497. IOS Press, 2016.

[5] William Bechtel and Adele Abrahamsen. Explanation: A mechanist alternative. *Studies in History and Philosophy of Biol and Biomed Science*, 36(2):421–441, 2005.

[6] Mathieu Beirlaen, Jesse Heyninck, and Christian Straßer. Structured argumentation with prioritized conditional obligations and permissions. *Journal of Logic and Computation*, page exy005, 2018.

[7] Mathieu Beirlaen and Christian Straßer. A structured argumentation framework for detaching conditional obligations. *CoRR*, abs/1606.00339, 2016.

[8] Trevor J. M. Bench-Capon and Frans Coenen. Isomorphism and legal knowledge based systems. *Artificial Intelligence and Law*, 1(1):65–86, 1992.

[9] Gustavo Adrian Bodanza, Fernando Tohmé, and Marcelo Auday. Collective argumentation: A survey of aggregation issues around argumentation frameworks. *Argument & Computation*, 8(1):1–34, 2017.

[10] G. Bongiovanni, G. Postema, A. Rotolo, G. Sartor, C. Valentini, and D Walton, editors. *Handbook of Legal Reasoning and Argumentation*. Springer, 2018.

[11] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artificial Inteligence*, 171(5-6):286–310, 2007.

[12] Martin Caminada and Mikolaj Podlaszewski. Grounded semantics as persuasion dialogue. In *Proc. of 4th Conf. on Computational Models of Argument*, pages 478–485. IOS Press, 2012.

[13] José Carmo and Andrew J. I. Jones. Deontic logic and contrary-to-duties. In *Handbook of Philosophical Logic: Volume 8*, pages 265–343. Springer, 2002.

[14] Federico Cerutti, Sarah A. Gaggl, Matthias Thimm, and Johannes P. Wallner. Foundations of implementations for formal argumentation. *IfCoLog Journal of Logics and their Applications*, 4(8):2623–2706, 2017.

[15] Günther Charwat, Wolfgang Dvořák, Sarah A. Gaggl, Johannes P. Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation – a survey. *Artificial Intelligence*, 220:28 – 63, 2015.

[16] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[17] Phan Minh Dung and Phan Minh Thang. Closure and consistency in logic-associated argumentation. *Journal of Artificial Intelligence Research*, 49:79–109, 2014.

[18] Phan Minh Dung, Francesca Toni, and Paolo Mancarella. Some design guidelines for practical argumentation systems. In *In Proc. of the 3rd Conf. on Computational Models of Argument*, pages 183–194. IOS Press, 2010.

[19] Lon L. Fuller. Positivism and fidelity to law: A reply to professor hart. *Harvard Law Review*, 71(4):630–672, 1958.

[20] Dov Gabbay, John Horty, Xavier Parent, Ron van der Meyden, and Leendert van der Torre, editors. *Handbook of Deontic Logic and Normative Systems*. College Publications, 2013.

[21] Guido Governatori, Michael J. Maher, David Billington, and Grigoris Antoniou. Argumentation semantics for defeasible logics. *Journal of Logic and Computation*, 14(5):675–702, 2004.

[22] Guido Governatori, Francesco Olivieri, Antonino Rotolo, and Simone Scannapieco. Computing strong and weak permissions in defeasible logic. *Journal of Philosophical Logic*, 42(6):799–829, 2013.

[23] Guido Governatori and Antonino Rotolo. BIO logical agents: Norms, beliefs, intentions in defeasible logic. *Autonomous Agents and Multi-Agent Systems*, 17(1):36–69, 2008.

[24] Guido Governatori, Antonino Rotolo, Régis Riveret, Monica Palmirani, and Giovanni Sartor. Variants of temporal defeasible logics for modelling norm modifications. In *Proc. of the 11th*

*Int. Conf. on Artificial Intelligence and Law*, pages 155–159. ACM, 2007.

[25] H. L. A. Hart. Positivism and the separation of law and morals. *Harvard Law Review*, 71(4):593–629, 1958.

[26] Carl G. Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, 15(2):135–175, 1948.

[27] Jörgen Jörgensen. Imperatives and logic. *Erkenntnis*, 7(1):288–296, 1937.

[28] Frank Keil. Explanation and understanding. *Annual review of psychology*, 57:227–254, 2005.

[29] Hans Kelsen. *Pure Theory of Law*. Translated from the 2nd (Revised and Enlarged) German Edition by Knight M. Berkeley: University of California Press, 1967.

[30] Ho-Pun Lam, Guido Governatori, and Régis Riveret. On ASPIC$^+$ and Defeasible Logic. In *Proc. of 6th Int. Conf. on Computational Models of Argument*, Amsterdam, 2016. IOS Press.

[31] Peter Lipton. What good is an explanation? In *Explanation: Theoretical Approaches and Applications*, pages 43–59. Springer, 2001.

[32] Tania Lombrozol. The structure and function of explanations. *Trends in Cognitive Sciences*, 10(10):464–470, 2006.

[33] David Makinson and Leendert van der Torre. Permission from an input/output perspective. *Journal of Philosophical Logic*, 32(4):391–416, 2003.

[34] David Makinson and Leendert W. N. van der Torre. Input/output logics. *Journal of Philosophical Logic*, 29(4):383–408, 2000.

[35] Paul McNamara. Deontic logic. In Dov M. Gabbay and John Woods, editors, *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of the History of Logic*, pages 197 – 288. North-Holland, 2006.

[36] Sanjay Modgil and Martin Caminada. Proof theories and algorithms for abstract argumentation frameworks. In *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.

[37] Sanjay Modgil and Henry Prakken. The *ASPIC*$^+$ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[38] Henry Prakken and Giovanni Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1-2):25–75, 1997.

[39] Henry Prakken and Giovanni Sartor. Formalising arguments about norms. In *Proc. of the 26th Annual Conf. on Legal Knowledge and Information Systems*, pages 121–130. IOS Press, 2013.

[40] Régis Riveret, Alexander Artikis, Jeremy V. Pitt, and Erivelton G. Nepomuceno. Self-governance by transfiguration: From learning to prescription changes. In *Proc. of 8th Int. Conf. on Self-Adaptive and Self-Organizing Systems*, pages 70–79. IEEE Computer Society, 2014.

[41] Régis Riveret, Pietro Baroni, Yang Gao, Guido Governatori, Antonino Rotolo, and Giovanni Sartor. A labelling framework for probabilistic argumentation. *Annals of Mathematics and Artificial Intelligence*, 83(1):21–71, 2018.

[42] Régis Riveret, Yang Gao, Guido Governatori, Antonino Rotolo, Jeremy Pitt, and Giovanni Sartor. A probabilistic argumentation framework for reinforcement learning agents - towards a mentalistic approach to agent profiles. *Autonomous Agents and Multi-Agent Systems*, 33(1-2):216–274, 2019.

[43] Régis Riveret and Guido Governatori. On learning attacks in probabilistic abstract argumentation. In *Proc. of the 15th Int. Conf. on Autonomous Agents & Multiagent Systems*, pages 653–661. ACM, 2016.

[44] Alf Ross. *Directives and Norms*. Humanities Press, 1967.

[45] Giovanni Sartor. *Legal Reasoning: A Cognitive Approach to the Law*. Springer, 2005.

[46] Audun Stolpe. A theory of permission based on the notion of derogation. *Journal of Applied Logic*, 8(1):97–113, 2010.

[47] Christian Straßer and Ofer Arieli. Normative reasoning by sequent-based argumentation. *Journal of Logic and Computation*, page exv050, 2015.

[48] Leendert W. N. van der Torre and Serena Villata. An ASPIC-based legal argumentation framework for deontic reasoning. In *Proc. of 5th Int. Conf. on Computational Models of Argument*, pages 421–432. IOS Press, 2014.

[49] Georg Henrik von Wright. *Norm and Action: A Logical Enquiry*. Routledge and Kegan Paul, 1963.

Received 2018