

Three Concepts of Defeasible Permission

Guido GOVERNATORI^{a,1}, Francesco OLIVIERI^{a,b,c}
Antonino ROTOLO^d and Simone SCANNAPIECO^{a,b,c}

^a*NICTA, Queensland Research Laboratory, Australia*²

^b*Department of Computer Science, University of Verona, Italy*

^c*Institute for Integrated and Intelligent Systems, Griffith University, Australia*

^d*CIRSFID, University of Bologna, Italy*

Abstract In this paper we propose an extension of Defeasible Logic to represent different concepts of defeasible permission. Special attention is paid in particular to permissive norms that work as exceptions to opposite obligations.

Keywords. Strong permission, Weak permission, Defeasible Logic

1. Introduction

The concept of permission plays an important role in the legal domain, since it may be crucial to characterise notions such as those of legal authorisation and derogation [15]. Despite that, it was not so extensively investigated in deontic logic as the notion of obligation. For a long time, deontic logicians mostly viewed permission as the dual of obligation: $Pa \equiv \neg O\neg a$. This view is unsatisfactory and has been criticised (see [2,1]).

One important distinction that traditionally contributed to a richer account of this concept is the one between weak (or negative) and strong (or positive) permission [19]. The former corresponds to saying that some a is permitted if $\neg a$ is not provable as mandatory. In other words, something is allowed by a (legal) code iff it is not prohibited by that code. At least when we deal with unconditional obligations, the notion of weak permission is trivially equivalent to the dual of obligation [12]. The concept of strong permission is more complicated, as it amounts to saying that some a is strongly permitted by a (legal) code if such a code explicitly states that a is permitted. The complexities of this concept depend on the fact that, besides “the items that a code explicitly pronounces to be permitted, there are others that in some sense follow from the explicit ones”. The problem is hence “to clarify the inference from one to the other” [12, p. 391–2].

Features such as the distinction between strong and weak permission show the multifaceted aspects of permissive norms. However, besides a few exceptions [12,5,6,7,17,16], most logicians still overlook this research issue. Nevertheless, some significant contributions have been offered by this bunch of literature. In particular, [12,5,6,17] raise and discuss the following points:

¹Corresponding author: guido.governatori@nicta.com.au.

²NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy, the Australian Research Council through the ICT Centre of Excellence program and the Queensland Government.

- despite some radical critiques [14,13], we should fruitfully keep the distinction between weak and strong permission;
- we may have different types of strong permissions (or, better, of permissions that logically follow from explicit permissive norms), according to whether
 - * we statically determine what is actually permitted given what is obligatory and what is explicitly permitted;
 - * we dynamically determine “the limits on what may be prohibited without violating static permissions” [5];
- especially in the *law*, strong permissions can play a role not only in overruling any incompatible prohibition, but also in stating exceptions to obligations [4];
- strong permissions make sense even when any incompatible prohibitions are not in the legal system; permissions have a dynamic behaviour and prevent that future prohibitions hold in general, or apply to specific contexts.

This paper moves from the above points with the purpose of studying the concepts of weak and strong permission within a new extension of Defeasible Logic (DL) [3]. There are several advantages of this choice:

1. DL is a computationally efficient logical framework able to capture various aspects of non-monotonic reasoning, and so of the defeasible character of permissive norms; in particular, we will see that strong permissions can be represented both introducing a new non-monotonic consequence relation for permission, or not;
2. The proposed extension of DL embeds operators able to express
 - (a) ordered sequences of contrary-to-duty obligations [9,10], in combination with
 - (b) ordered sequences of strong permissions which are supposed to derogate or make exceptions to prohibitions; this is a specific novelty of this contribution, as sequences of permissions allow us to represent preferences between permissions (i.e., exceptions) which are not necessarily incompatible with each other.

The layout of the paper is as follows. Section 2 informally discusses how to characterise permissions in DL. Section 3 technically presents the machinery and states some results. Section 4 discusses some related work and provides a summary of the paper.

2. Three Concepts of Permission

Let us first summarise some preliminary intuitions behind our logical framework.

1. Permissive and prescriptive norms are represented by means of defeasible rules, whose conclusions normally follow unless they are defeated by contrary evidence. For example, the rule $Order \Rightarrow_O Pay$ says that, if we send a purchase order, then we will be defeasibly obliged to pay; the rule $Order, Creditor \Rightarrow_P \neg Pay$ states that if we send an order, we are normally not obliged to pay if we are creditors towards the vendor for the same amount we have to pay for that order.
2. Rules introduce modalities: if we have $a \Rightarrow_O b$ and a , then we obtain Ob .
3. For the sake of simplicity, modal literals can only occur in the antecedent of rules. This is in line with our idea that the applicability of rules labeled with modality X is the condition for deriving literals modalised with X . In other words, we do not admit rules such as $a_1, \dots, a_n \Rightarrow_O Pb$.

4. Legal norms often specify mandatory actions to be taken in case of their violation. In general, obligations in force after some other obligations have been violated correspond to contrary-to-duty (CTD) obligations. These constructions affect the formal characterisation of compliance since they identify situations that are not ideal, but still acceptable. A compact representation of CTDs may resort to the non-boolean connective \otimes [9]: a formula like $x \Rightarrow_O a \otimes b$ means that, if x is the case, then a is obligatory, but if the obligation a is not fulfilled, then the obligation b is activated and becomes in force until it is satisfied or violated.

Notice that O and P are not simple labels: they are modalities. O is non-reflexive³: if a rule $b_1, \dots, b_n \Rightarrow_O a$ is applicable, no contrary evidence defeats it, and we know that $\neg a$ is the case, then we do not obtain a conflict between the fact $\neg a$ and the conclusion of the rule, since such a conclusion is Oa ; $\neg a$ consists rather in violation of Oa . On the other hand, the modality P is logically characterized in such a way as two rules for P supporting a and $\neg a$ do not clash, but a rule $\Rightarrow_P b$ attacks a rule $\Rightarrow_O \neg b$ (and vice versa).

This basic framework already allows us to express three types of permission and capture different aspects of how defeasibility affects them [18]. Indeed, like standard DL, our extension is able to establish the relative strength of any rule (hence, to solve rule conflicts) and has two types of attackable rules: defeasible rules and defeaters. Defeaters in DL are a special kind of rule: they are used to prevent conclusions but not to support them. For example, *SpecialOrder, PremiumCustomer $\rightsquigarrow_O \neg PayBy7Days$* states that premium customers placing special orders might be exempt from paying by 7 days: it can prevent the derivation of an obligation to pay within the deadline, but it cannot be used to directly derive any conclusion.

Weak Permission A first way to define permissions in DL is by simply considering weak permissions and stating that the opposite of what is permitted is not provable as obligatory. Consider a normative system consisting of just the following two rules:

$$r_1 : Park, Vehicle \Rightarrow_O \neg Enter \quad r_2 : Park, Emergency \Rightarrow_O Enter$$

Here the normative system does not contain any permissive norm. However, since DL is a skeptical non-monotonic logic, in case both r_1 and r_2 could fire we cannot conclude either that it is prohibited to enter nor that it obligatory, because we do not know what rule is stronger. Hence, in this context, both $\neg Enter$ and $Enter$ are weakly permitted.

This is the most direct way to define the idea of weak permission: something is permitted by a code iff it is not prohibited by that code. In the formal language, this possibility consists in adding the modality P without having specific rules for it. Indeed, we can have modal literals, such as Pq in the antecedent of a rule (which can be obtained by showing that no opposite obligation is provable).

Explicit Permissions Are Defeaters. Any defeasible rule supporting some Oa can lead to infer this obligation and also block another obligation $O\neg a$. In this sense, this type of rule is not fully satisfactory to investigate the concept of permission, since it can only be used to characterize the notion of weak permission. Thus, there are good reasons to argue that defeaters for O are suitable to express an idea of strong permission [11]. *Explicit* rules such as $r : a \rightsquigarrow_O q$ state that a is a specific reason for *blocking* the derivation of $O\neg q$ (but not for proving something): this rule does not support any conclusion but states that q is *not undesirable* from the deontic perspective. Consider this example:

³As is well-known, in a non-reflexive modal logic a does not follow from Xa , where X is a modal operator.

$$r_1 : \text{Weekend, AirPollution} \Rightarrow_O \neg \text{UseCar} \quad r_2 : \text{Weekend, Emergency} \rightsquigarrow_O \text{UseCar}$$

Rule r_1 states that on weekends it is forbidden to use private cars if certain air pollution limit values are exceeded. Defeater r_2 is in fact an exception to r_1 , and so it seems to capture the above recalled idea that explicit permissive norms (especially in the law) provide exceptions to obligations.

Using Permissive Rules Another approach is based on introducing specific rules for deriving permissions [12,5]. Let us consider the following scenario:

$$r_1 : \text{Weekend, AirPollution} \Rightarrow_O \neg \text{UseCar} \quad r'_2 : \text{Weekend, Emergency} \Rightarrow_P \text{UseCar}$$

As r_2 in the previous scenario, r'_2 looks like an exception to r_1 . The apparent difference between r_2 and r'_2 is that the latter is directly used to prove $P\text{UseCar}$. The question is: does it amount to a real difference?

Indeed, also r_2 , although it is a defeater, is specifically used to derive it. In addition, rules like r'_2 do not attack other permissive rules, but are in conflict only with rules for obligation that allow to prove the opposite conclusion of the permission we want to derive. This precisely holds for defeaters.

Moreover, let us suppose to have a defeater $s : a \rightsquigarrow_P b$. Does s attack a rule like $\Rightarrow_P \neg b$? If this was the case, s would be close to an obligation (Pb does not attack $P\neg b$), thus making useless to introduce defeaters for P. But, if this is not the case, s can only attack $\Rightarrow_O \neg b$, thus being equivalent with $s' : a \rightsquigarrow_O b$.

Hence, although it is admissible to have defeaters, we do not need to distinguish defeaters for O from those for P.

One significant difference between \rightsquigarrow and \Rightarrow_P is that only the latter rule type can meaningfully express preference orders among different permissions which are supposed to explicitly derogate or make exceptions to prohibitions. To do so, the formal language can be enriched as follow:

5. What we have done at point 4 above with the operator \otimes can be extended to permissive rules with the subscripted arrow \Rightarrow_P . In other words, we can introduce a new non-boolean connective \odot for sequences of permissions. As done with \otimes , given a rule $r : \Rightarrow_P a \odot b$, we can proceed through the \odot -chain to obtain the derivation of Pb . However, strictly speaking, permissions cannot be violated and does not make sense that, given $\neg a$, we get Pb from r . Rather, the reason to proceed here in the chain is that the normative system allows us to show that $O\neg a$ is provable or at least defeats rule r , the one supporting Pa . Hence, \odot still establishes a preference order among strong permissions and, in case opposite obligations at least defeat the first n permissions in the chain, the subsequent $n + 1$ th permission holds. This is significant especially when strong permissions are exceptions to obligations.

Consider the following scenario:

$$\text{Facts} = \{\neg \text{CallFiremen}, \neg \text{CallAmbulance}, \text{Fire}, \text{CarCrash}, \text{Injured}\}$$

$$\text{Rules} = \{r_1 : \text{CarCrash} \Rightarrow_O \text{CallAmbulance} \otimes \text{Help}$$

$$r_2 : \text{Fire} \Rightarrow_O \text{CallFiremen} \otimes \text{Extinguish}$$

$$r_3 : \text{Fire}, \text{CarCrash}, \text{Injured}, \neg \text{CallAmbulance} \Rightarrow_P \neg \text{Help} \odot \neg \text{Extinguish}\}$$

$$\text{Priorities} = r_3 \text{ is stronger than } r_2 \text{ and weaker than } r_1$$

Rule r_1 says that, if you are in a close proximity of a car crash, you are under the obligation to call an ambulance; if you do not do it for some reason, you still have the obligation to help. Rule r_2 states that, in case of fire, you have the obligation to call fire brigades, but if you do not do it you have to try to extinguish the fire. Finally, rule r_3 provides an order between two exceptions in a case where you are in proximity of a fire, of a car crash, you are injured and have not called an ambulance. All rules are triggered. r_1 leads to obtain *OCallAmbulance*, which is violated by a fact, so the obligation to help follows. The primary obligation of r_2 is violated, but r_3 is applicable, too, and is stronger than r_2 : this leads to derive $P \rightarrow \text{Extinguish}$, so there is no way to compensate the violation of *OCallFiremen*, since r_3 provides a secondary exception.

The next section presents in detail the logical framework able to express the three types of defeasible permissions that we have informally discussed so far: weak defeasible permission, strong defeater-based permission, and strong permission admitting preference orders.

3. Defeasible Deontic Logic with Strong Permission

In this section we first introduce the language adopted to formalise obligation and (strong) permissions in DL, and then we introduce the inferential mechanism in form of proof conditions defining the logic. Finally we show that the proposed formalisation enjoys properties suitable for the modelling of the notion of strong permission.

Let PROP be a set of propositional atoms, $\text{MOD} = \{O, P\}$ the set of modal operators, and Lab be a set of arbitrary labels. The set $\text{Lit} = \text{PROP} \cup \{\neg p \mid p \in \text{PROP}\}$ denotes the set of *literals*. The *complementary* of a literal q is denoted by $\sim q$; if q is a positive literal p , then $\sim q$ is $\neg p$, and if q is a negative literal $\neg p$, then $\sim q$ is p . The set of modal literals is $\text{ModLit} = \{Xl, \neg Xl \mid l \in \text{Lit}, X \in \text{MOD}\}$.

We introduce two preference operators, \otimes for obligations, and \odot for permissions, and we will use \oslash when we refer to one of them generically. These operators are used to build chains of preferences, called \oslash -expressions.

The formation rules for well-formed \oslash -expressions are: (a) every literal is an \oslash -expression; (b) if A is an \otimes -expression, B is an \odot -expression and c_1, \dots, c_k are literals then $A \otimes c_1 \otimes \dots \otimes c_k$ is an \otimes -expression, $B \odot c_1 \odot \dots \odot c_k$ is an \odot -expression, $A \odot B$ is an \oslash -expression; (c) every \otimes -expression and \odot -expression is an \oslash -expression; (d) nothing else is an \oslash -expression. We adopt the standard DL definitions of *strict rules*, *defeasible rules*, and *defeaters* [3]. However, for the sake of simplicity, and to better focus on the non-monotonic aspects that DL offers, in the remainder we use only defeasible rules and defeaters. In addition, we have to take the modal operators into account. Every rule is of the type $r : a_1, \dots, a_n \hookrightarrow C$, where

1. $r \in \text{Lab}$ is the name of the rule;
2. a_1, \dots, a_n , the *antecedent* of the rule, is the set of the premises of the rule (alternatively, it can be understood as the conjunction of all the literals in it). Each a_i is either a literal or a modal literal;
3. $\hookrightarrow \in \{\Rightarrow_X, \rightsquigarrow\}$ denotes the type of the rule. If \hookrightarrow is \Rightarrow_X , the rule is a *defeasible rule*, while if \hookrightarrow is \rightsquigarrow , the rule is a *defeater*. The subscript $X \in \text{MOD}$ in defeasible rules represents the modality introduced by the rule itself. The mode of a rule tells us what kind of conclusion we can obtain from the rule; as we argued, we do not need to label \rightsquigarrow with any modality.

4. C is the *consequent* (or *head*) of the rule, which is an \odot -expression. Two constraints apply on the consequent of a rule: (a) if \leftrightarrow is \rightsquigarrow , then C is a single literal; (b) if $X = P$, then C must be an \odot -expression.

Given an \odot -expression A , the *length* of A is the number of elements in it. Given an \odot -expression $A \odot b$, the *index* of b is n iff the length of $A \odot b$ is n . We also say that b appears at index n in $A \odot b$. $R[a, k]$ is the set of rules where element a is at index k in the head of the rules.

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

- R_{def} denotes the set of all defeaters in the set R ;
- $R[q, n]$ is the set of rules where q appears at index n in the consequent;
- $R^O[q, n]$ is the set of (defeasible) rules where q appears at index n and the operator at index $n - 1$ is \otimes ; the set of (defeasible) rules where q appears at any index n satisfying the above constraints is denoted by $R^O[q]$;
- similarly $R^P[q, n]$ is the set of rules where q appears at index n , and the operator at index $n - 1$ is \odot ; the set of (defeasible) rules where q appears at any index n satisfying the above constraints is denoted by $R^P[q]$.

A *Defeasible Theory* is a structure (F, R, \succ) , where F , the set of facts, is a set of literals and modal literals, R is a set of rules and \succ , the superiority relation, is a binary relation over R .

A theory corresponds to a normative system, i.e., a set of norms, where every norm is modelled by rules. The superiority relation is used for conflicting rules, i.e., rules whose conclusions are complementary literals, in case both rules fire. Notice that we do not impose any restriction on the superiority relation: it is just a binary relation determining the relative strength of two rules.

Proofs in a defeasible theory T are linear derivations, i.e., sequences of *tagged literals* in the form of $+\partial_X q$ and $-\partial_X q$. Given $X \in \text{MOD}$, $+\partial_X q$ means that q is defeasibly provable in T with modality X , while $-\partial_X q$ means that q is defeasibly refuted. The initial part of length i of a proof P is denoted by $P(1..i)$.

The first thing to do is to define when a rule is applicable or discarded. A rule is *applicable* for a literal q if all non-modalised literals in the antecedent are given as facts and all the modalised literals have been proved (with the appropriate modalities). On the other hand, a rule is *discarded* if at least one of the modalised literals in the antecedent has not been proved (or is not a fact in the case of non-modalised literals). However, as literal q could not appear as the first element in an \odot -expression in the head of the rule, some additional conditions on the consequent of rules must be satisfied.

Definition 1. A rule $r \in R[q, j]$ such that $C(r) = c_1 \otimes \dots \otimes c_{l-1} \odot c_l \odot \dots \odot c_n$ is applicable for literal q at index j , with $1 \leq j < l$, in the condition for $\pm\partial_O$ iff

(1) for all $a_i \in A(r)$:

- (1.1) if $a_i = Ol$ then $+\partial_O l \in P(1..n)$;
- (1.2) if $a_i = \neg Ol$ then $-\partial_O l \in P(1..n)$;
- (1.3) if $a_i = Pl$ then $+\partial_P l \in P(1..n)$;
- (1.4) if $a_i = \neg Pl$ then $-\partial_P l \in P(1..n)$;
- (1.5) if $a_i = l \in \text{Lit}$ then $l \in F$, and

(2) for all $c_k \in C(r)$, $1 \leq k < j$, $+\partial_O c_k \in P(1..n)$ and $(c_k \notin F \text{ or } \sim c_k \in F)$.

Conditions (1.1)–(1.5) represent the above requirements; condition (2) on the head of the rule states that each element c_k prior to q must be derived as an obligation, and a violation of such obligation has occurred.

Definition 2. A rule $r \in R[q, j]$ such that $C(r) = c_1 \otimes \cdots \otimes c_{l-1} \odot c_l \odot \cdots \odot c_n$ is applicable for literal q at index j , with $l \leq j \leq n$ in the condition for $\pm \partial_P$ iff

- (1) for all $a_i \in A(r)$:
 - (1.1) if $a_i = Ol$ then $+\partial_O l \in P(1..n)$;
 - (1.2) if $a_i = \neg Ol$ then $-\partial_O l \in P(1..n)$;
 - (1.3) if $a_i = Pl$ then $+\partial_P l \in P(1..n)$;
 - (1.4) if $a_i = \neg Pl$ then $-\partial_P l \in P(1..n)$;
 - (1.5) if $a_i = l \in \text{Lit}$ then $l \in F$, and
- (2) for all $c_k \in C(r)$, $1 \leq k < l$, $+\partial_O c_k \in P(1..n)$ and $(c_k \notin F \text{ or } \sim c_k \in F)$, and
- (3) for all $c_k \in C(r)$, $l \leq k < j$, $-\partial_P c_k \in P(1..n)$.

The only difference with respect to $\pm \partial_O$ is the presence of an additional condition, stating that all permissions prior to q must be refuted (condition (3)).

Definition 3. A rule $r \in R[q, j]$ such that $C(r) = c_1 \otimes \cdots \otimes c_{l-1} \odot c_l \odot \cdots \odot c_n$ is discarded for literal q at index j , with $1 \leq j \leq n$ in the condition for $\pm \partial_O$ or $\pm \partial_P$ iff

- (1) there exists $a_i \in A(r)$ such that
 - (1.1) if $a_i = Ol$ then $-\partial_O l \in P(1..n)$;
 - (1.2) if $a_i = \neg Ol$ then $+\partial_O l \in P(1..n)$;
 - (1.3) if $a_i = Pl$ then $-\partial_P l \in P(1..n)$;
 - (1.4) if $a_i = \neg Pl$ then $+\partial_P l \in P(1..n)$;
 - (1.5) if $a_i = l \in \text{Lit}$ then $l \notin F$, or
- (2) there exists $c_k \in C(r)$, $1 \leq k < l$, such that either $-\partial_O c_k \in P(1..n)$ or $c_k \in F$, or
- (3) there exists $c_k \in C(r)$, $l \leq k < j$, such that $+\partial_P c_k \in P(1..n)$.

In this case, condition (2) ensures that an obligation prior to q in the chain is not in force or has already been fulfilled (thus, no reparation is required), while condition (3) states that there exists at least one explicit derived permission prior to q .

We now introduce the proof conditions for $\pm \partial_O$ and $\pm \partial_P$.

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

- (1) $Oq \in F$ or
 - (2.1) $O \sim q \notin F$ and $\neg Oq \notin F$ and $P \sim q \notin F$ and
 - (2.2) $\exists r \in R^O[q, i]$ such that r is applicable for q , and
 - (2.3) $\forall s \in R[\sim q, j]$, either
 - (2.3.1) s is discarded, or either
 - (2.3.2) $s \in R^O$ and $\exists t \in R[q, k]$ such that t is applicable for q and $t \succ s$, or
 - (2.3.3) $s \in R^P \cup R_{def}$ and $\exists t \in R^O[q, k]$ such that t is applicable for q and $t \succ s$.

To show that q is defeasibly provable as an obligation, there are two ways: (1) the obligation of q is a fact, or (2) q must be derived by the rules of the theory. In the second case, three conditions must hold: (2.1) $\sim q$ is not provable as an obligation using the set of modalised facts at hand; (2.2) there must be a rule introducing obligation for q which can apply; (2.3) every rule s for $\sim q$ is discarded or defeated by a stronger rule for q . If s is an obligation rule, then it can be counterattacked by any type of rule; if s is a defeater or a permission rule, then only an obligation rule can counterattack it.

Here below are the conditions for $-\partial_O$ and $\pm\partial_P$:

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

(1) $Oq \notin F$ and either

(2.1) $O\sim q \in F$ or $\neg Oq \in F$ or $P\sim q \in F$ or

(2.2) $\forall r \in R^O[q, i]$ either r is discarded for q , or

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

(2.3.1) s is applicable for $\sim q$, and

(2.3.2) if $s \in R^O$ then $\forall t \in R[q, k]$, either t is discarded or $t \not\prec s$, and

(2.3.3) if $s \in R^P \cup R_{def}$ then $\forall t \in R^O[q, k]$, either t is discarded or $t \not\prec s$.

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

(1) $Pq \in F$ or

(2.1) $O\sim q \notin F$ and $\neg Pq \notin F$ and

(2.2) $\exists r \in R^P[q, i]$ such that r is applicable for q , and

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

(2.3.1) s is discarded for $\sim q$, or

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

This last proof condition differs from its counterpart for obligation in two aspects: we allow scenarios where both $+\partial_P q$ and $+\partial_P \sim q$ hold, but $+\partial_O \sim q$ must not hold (2.1); any applicable rule s supporting $\sim q$ can be counterattacked by any type of rule t supporting q , as s must be an obligation rule, and permission rules can only be attacked by obligation rules (2.3).

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

(1) $Pq \notin F$ and either

(2.1) $O\sim q \in F$ or $\neg Pq \in F$, or

(2.2) $\forall r \in R^P[q, i]$, either r is discarded, or

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

(2.3.1) s is applicable for $\sim q$, and

(2.3.2) $\forall t \in R[q, k]$, either t is discarded or $t \not\prec s$.

The logic resulting from the above proof conditions enjoys properties describing the appropriate behaviour of the modal operators. A Defeasible Theory D is *consistent* iff F does not contain pairs of complementary (modal) literals, thus if D does not contain pairs like Ol and $\sim Ol$, Pl and $\sim Pl$, and l and $\sim l$. A Defeasible Theory D is *O-consistent* iff for any literal l , F does not contain any of the following pairs: Ol and $O\sim l$, Ol and $P\sim l$.

As usual, given a Defeasible Theory D we will use $D \vdash \pm\partial_{\square} p$ iff there is a derivation of $\pm\partial_{\square} p$ from D .

Proposition 1. *Let D be a consistent Defeasible Theory, and $\square \in \{O, P\}$. For any literal l , it is not possible to have both $D \vdash +\partial_{\square} l$ and $D \vdash -\partial_{\square} l$.*

The meaning of the above proposition is that it not possible to prove that a literal is at the same time obligatory and not obligatory, and permitted and not permitted.

Proposition 2. *Let D be an O-consistent Defeasible Theory, then for any literal l , it is not possible to have both $D \vdash +\partial_O l$ and $D \vdash +\partial_O \sim l$.*

The meaning of the proposition is that no formula is both obligatory and forbidden at the same time. However, the proposition does not hold for permission. It is possible to have both the explicit permission of l and the explicit permission of $\sim l$.

The basic relationships between permissions and obligations are governed by the following proposition:

Proposition 3. *Let D be an O -consistent Defeasible Theory. For any literal l :*

1. *if $D \vdash +\partial_O l$, then $D \vdash -\partial_O \sim l$;*
2. *if $D \vdash +\partial_P l$, then $D \vdash -\partial_P \sim l$;*
3. *if $D \vdash +\partial_P l$, then $D \vdash -\partial_O \sim l$.*

The combination of the two items above describes the consistency between obligation and permission. Part 3 also gives the relationships between strong and weak permission. As we discussed in Section 1, a weak permission is a permission obtained from the failure to derive the opposite obligation. This means that we have the weak permission of p , when we have $-\partial_O \sim p$, and Proposition 3 Part 3 guarantees that we have it when we have $+\partial_P p$.

Let us see how the logic works with the example introduced at the end of Section 1.

Example. Let us recall the scenario reported at the end of Section 2, and formally explain the conclusions of the theory using applicability of rules and proof tags as defined above. The primary obligation to call the ambulance is obtained (i.e. we derive $+\partial_O \text{CallAmbulance}$), but the obligation is violated as $\neg \text{CallAmbulance} \in F$, making r_1 applicable for *Help*; rule r_3 is applicable for literal $\neg \text{Help}$ and could attack r_1 , but $r_1 \succ r_3$, thus we have also $+\partial_O \text{Help}$. Also *CallFiremen* is derived as an obligation but it is violated, thus rule r_2 is applicable for literal *Extinguish* for $+\partial_O$. As $+\partial_O \text{Help}$ holds, r_3 is applicable for $\neg \text{Extinguish}$, and $r_3 \succ r_2$, thus we derive $+\partial_P \neg \text{Extinguish}$.

4. Summary and Related Work

In this paper we proposed an extension of Defeasible Logic (DL) to represent three concepts of defeasible permission. In particular, we have discussed different types of explicit permissive norms that work as exceptions to opposite obligations. Also, we considered how strong permissions can be represented both with and without introducing a new consequence relation for inferring conclusions from explicit permissive norms. Finally, we combined a preference operator applicable to contrary-to-duty obligations with a new one representing ordered sequences of strong permissions which derogate to prohibitions. The contribution is new as compared to our previous work (such as [9,10,8]), since (1) it systematically discusses different concepts of permissions in DL, which were only briskly introduced in [8] to discuss their interplay with intentions and beliefs in MAS, and (2) it proposes a new concept of permission admitting preference orderings, which is not considered in any of our earlier contributions on preference deontic logics [9,10].

The works in the literature that are closer to our approach are [12,5,17] which, however, are all based on Input/Output logic. It is difficult to compare in detail that formalism with DL, but there are some general similarities: those works (a) model weak and strong permissions by distinguishing in an analogous way a consequence relation for obligation and one for permission; (b) distinguish, in a way similar to what we do, between

permissions rebutting obligations and permissions providing exceptions; (c) distinguish between static and dynamic permissions: in particular, the former concept of [12,5] can be directly expressed in our framework, while the latter can be captured but in a different way, due to the skeptical character of DL (see [5, Definition 2]). We believe, in particular, that a study of the relation between our approach and the one in [12,5] can shed light on more specific properties that the just presented concepts of permission enjoy: space reasons do not allow us to address this issue, which is left to future research.

Although the introduction of the new operator \odot to express preferences between explicit permissions is a novelty of this paper, a somehow similar idea has been suggested (though with different purposes) by [5], where a preference relation among generators (for obligations and permissions) was introduced. Technically, it is not clear if that approach can be reframed in our setting. In fact, adopting that option in DL would not work, as the superiority relation in DL plays a role in the proof theory only in case of rule conflicts. A clear advantage of the current proposal is anyway that it adopts a richer formal language with modal operators, which can occur in the applicability conditions of rules.

References

- [1] C.E. Alchourrón. Philosophical foundations of deontic logic and the logic of defeasible conditionals. In J. J. Meyer and R. J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*. Wiley, 1993.
- [2] C.E. Alchourrón and E. Bulygin. Permission and permissive norms. In W. Krawietz et al., editor, *Theorie der Normen*. Duncker & Humblot, 1984.
- [3] G. Antoniou, D. Billington, G. Governatori, and M.J. Maher. Representation results for defeasible logic. *ACM Trans. Comput. Logic*, 2:255–287, 2001.
- [4] N. Bobbio. *Teoria della norma giuridica*. Giappichelli, 1958.
- [5] G. Boella and L. van der Torre. Permissions and obligations in hierarchical normative systems. In *Proceedings of ICAIL 2003*. ACM Press, 2003.
- [6] G. Boella and L. van der Torre. Permissions and undercutters. In *Proceedings of the NRAC'03*, Acapulco, 2003.
- [7] M.A. Brown. Conditional obligation and positive permission for agents in time. *Nordic Journal of Philosophical Logic*, 5(2):83–111, 2000.
- [8] G. Governatori, V. Padmanabhan, A. Rotolo, and A. Sattar. A defeasible logic for modelling policy-based intentions and motivational attitudes. *Logic Journal of the IGPL*, 17(3):227–265, 2009.
- [9] G. Governatori and A. Rotolo. Logic of violations: A Gentzen system for reasoning with contrary-to-duty obligations. *Australasian Journal of Logic*, 4:193–215, 2006.
- [10] G. Governatori and A. Rotolo. Justice delayed is justice denied: Logics for a temporal account of reparations and legal compliance. In *Proceedings of CLIMA XII*. Springer, 2011.
- [11] G. Governatori, A. Rotolo, and G. Sartor. Temporalised normative positions in defeasible logic. In *Proceedings of ICAIL 2005*. ACM Press, 2005.
- [12] D. Makinson and L. van der Torre. Permission from an input/output perspective. *Journal of Philosophical Logic*, 32(4):391–416, 2003.
- [13] K. Opalek and J. Wolenski. Normative systems, permission and deontic logic. *Ratio Juris*, 4, 1991.
- [14] A. Ross. *Directives and norms*. Routledge and Kegan Paul, 1968.
- [15] G. Sartor. *Legal Reasoning: A Cognitive Approach to the Law*. Springer, 2005.
- [16] A. Stolpe. Relevance, derogation and permission. In *Proceedings of DEON 2010*. Springer, 2010.
- [17] A. Stolpe. A theory of permission based on the notion of derogation. *J. Applied Logic*, 8(1):97–113, 2010.
- [18] L. van der Torre and Y.-H. Tan. The many faces of defeasibility in defeasible deontic logic. In D. Nute, editor, *Defeasible Deontic Logic*. Kluwer, 1997.
- [19] G.H. Wright. *Norm and action: A logical inquiry*. Routledge and Kegan Paul, 1963.