

Permissions and Obligations in Hierarchical Normative Systems

Guido Boella
Dipartimento di Informatica
Università di Torino
Italy
guido@di.unito.it

Leendert van der Torre
SEN-3
CWI Amsterdam
The Netherlands
torre@cwi.nl

ABSTRACT

In this paper we discuss different types of permissions and their roles in deontic logic. We study the distinction between weak and strong permissions in the context of input/output logic, combining the logic with constraints, priorities and hierarchies of normative authorities. In this setting we observe that the notion of prohibition immunity no longer applies, and we introduce a new notion of permission as exception and a new distinction between static and dynamic norms. We show that strong permissions can dynamically change a normative system by adding exceptions to obligations, provide an explicit representation of what is permitted to the subjects of the normative system and allow higher level authorities to limit the changes that lower level authorities can do to the normative system.

1. INTRODUCTION

One of the reasons why deontic logic and legal studies are kept separate is the notorious problem of the definition of *permissions* [24]. In particular, there is no agreement on the need of a notion of permission distinct from the notion of obligation. Moreover, it is still disputable which are the types of permissions and which roles permissions can play in a normative system. We believe that these problems can be overcome by extending recent approaches to deontic logic as, e.g., input/output logic [21].

Starting from the analysis of strong permissions given by Bulygin [13], in this paper we address the following questions:

- Why should an explicit notion of permission be distinguished from the mere absence of obligation?
- What is the meaning of permissions for the citizens? Are citizens helped by the presence of permissive norms to understand the limits of the spheres of what is permitted and of what is forbidden?

- How do permissive norms help legislative authorities which build normative systems? Is it possible that a normative system changes dynamically without permissions which work as exceptions to obligations?
- It is known that a normative system based on permissions only is useless [19]. But which characteristics must a normative system have for permissions to be fully useful? Is a flat normative system with a single authority sufficient? Are multiple authorities needed? Should they be arranged in a hierarchical structure?

The method we use to answer these questions is an extension of the approach of Makinson and van der Torre [20]. They propose input/output logic to formalize conditional norms, obligations, goals, *et cetera*. They extend the logic to formalize negative, static and dynamic permissions [22], and they introduce constraints to deal with contrary-to-duty reasoning [21]. We first consider the combination of permissions and constraints in input/output logic and observe the following two phenomena:

- The notion of dynamic permission no longer corresponds to so-called prohibition immunity,
- The notion of permission as exception has not been covered.

We introduce priorities and hierarchies of normative systems in input/output logic. If a normative system is structured in a hierarchical way, there is the possibility that norms from different authorities conflict with each other. Moreover, authorities may change the normative status only of certain behaviors they are competent of. We also consider the strategies used to determine which norms are valid in a certain situation using *meta-norms*.

The layout of this paper is as follows. In Section 2 we present the different positions of law scholars about the existence and the functions of permissions in legal systems. In Section 3 we consider the limits of input/output logic with constraints for dealing with permissions. In Section 4 we introduce permissions as exceptions using conflict resolution and in Section 5 we make an alternative distinction between static and dynamic norms using hierarchies of normative systems, discussing the notion of competence and of redundancy of norms. Related work and summary close the paper.

2. DEFINITIONS OF PERMISSIONS

In Section 2.1 we present the distinction between weak and strong permissions introduced by von Wright. However, von Wright does not motivate adequately the role of strong permissions. In Section 2.2 we consider the position of legal scholars who argue that the function of permissions is to derogate to obligations. In Section 2.3 we contrast the position which claims that since permissions derogate to obligations they always presuppose imperative norms and must be defined in function of them. In Section 2.4 we present Bulygin's position which we adopt in this paper: he argues that permissions show their full meaning only in the context of a hierarchical normative system composed of more than one normative authority. Finally, in Section 2.5 we discuss the role of meta-norms.

2.1 Permissions are weak or strong?

The first question is whether permission is an autonomous normative category. The question arises because of the interdefinability of the deontic operators of permission and obligation: $P(q)$ iff $\neg O(\neg q)$; is permission only the mere absence of obligation or something which can be positively expressed by a norm?

According to von Wright there are two types of permissions, *weak permissions* and *strong permissions*: "An act will be said to be permitted in the weak sense if it is not forbidden; and it will be said to be permitted in the strong sense if it is not forbidden but subject to norm. [...] Weak permission is not an independent norm-character. Weak permissions are not prescriptions or norms at all. Strong permission only is a norm-character." [33, p.86].

In [32] the distinction is motivated on the basis of the distinction between weak and strong negation: weak norms consist in the mere absence of the contrary norms. Later, von Wright [33] argues that the strong permissions provide an answer to the need of disciplining new kinds of behaviors, filling gaps in law and needed to express competence laws. Where by competence he means the faculty to change the normative status of certain actions.

In particular, von Wright [33] says that "a higher order permission is to the effect that a certain authority *may* issue norms of a certain content. It is, we could say, a norm concerning the competence of a certain authority of norms", p.192. In contrast with other norms, such competence norms are characterized by the principle "whatever is not permitted is forbidden".

These last claims, however, are objectionable according to Ross: "the constitutional guarantee of certain freedoms [...] is a restriction of the power of the legislator, a disability which corresponds to an immunity on the part of the citizen" [29, p.125], rather than an imperative or permissive norm on competence. While gaps are better defined as constitutional rights which are not protected by laws so that they do not find a full accomplishment.

So, according to Ross, von Wright fails to justify the need for a notion of permission separate from that of obligation.

2.2 Permissions as exceptions

The distinction between weak and strong permissions is motivated in a different way by legal scholars: they individuate the main role of permissive norms in specifying exceptions to obligations. Bobbio [5] explains that "the difference between weak and strong permission becomes clear when

we think about the function of permissive norms. Permissive norms are subsidiary norms: subsidiary in that their existence presupposes the existence of imperative norms [...] a permissive norm is necessary when we have to repeal a preceding imperative norm or to derogate to it. That is to abolish a part of it (that in this case it is not necessary pre-existing because a law itself may prescribe a limit to its own extension)", p. 891-892.¹

Moreover, Bobbio [4] discusses the temporal relations existing between permissions and obligations: "the function of permissive norms is to abolish an imperative in certain circumstances or with reference to certain persons [...] permissive norms are distinguished according to the fact that they abolish an *imperative which precedes in time* and in this case are called *abolishing norms*, or a *contemporary imperative*, and in this case they generally have the function of *derogating norms*."

So for Bobbio:

- Permissions are conditional: they act on "certain circumstances or with reference to certain persons".
- Permissions presuppose the existence of imperative norms.
- Permissions are exceptions to those norms: they can cancel an imperative *in toto* or, given their conditional character, only in part.
- Permissions temporally follow imperative norms or may be issued at the same time as an obligation.

However, the use of permissions does not seem to be unavoidable. There could be no need to use them to derogate an obligation. At first sight, to express the derogation of an existing obligation could be sufficient to substitute it with a more complex conditional obligation. This obligation would rule out the exceptional circumstances specified in the permissions which should derogate it.

2.3 Do permissions presuppose obligations ?

The notion of permission is not enough to build a normative system, as Lewis [19]'s "master and slave" game has shown: for only obligation norms divide the possible actions into two categories or spheres: the sphere of prohibited actions and the sphere of permitted (i.e., not forbidden) actions or "the sphere of permissibility".

Ross [29] supports the view that without the context of an obligation a permission is not useful: "telling me what I am permitted to do provides no guide to conduct unless the permission is taken as an exception to a norm of obligation (which may be the general maxim that what is not permitted is prohibited). Norms of permission have the normative function only of indicating, within some system, what are the exceptions from the norms of the obligation of the system", p.120. His conclusion is very strong: "I know of no permissive legal rule which is not logically an exemption modifying some prohibition, and interpretable as the negation of an obligation", p.122.

For the role of permissions to indicate exceptions to an obligation they presuppose, it may be interesting to compare permissions with other mechanisms studied in, for example, nonmonotonic logic to reason with exceptions. One

¹Authors' translation from Italian text.

goal to introduce rules with exceptions in the representation, is that it is often inefficient to either list all exceptions when the rule is created, or update the rule each time a new exception comes up. Another goal, which has not been reached yet, is that reasoning with rules which allow exceptions may be more efficient due to the gain of reasoning by jumping to conclusions. In most approaches to nonmonotonic logic a default can only be defeated by another default, i.e., in our case an obligation can only be defeated by another obligation. In such approaches the focus is on studying different ways to resolve conflicts between rules, for example based on the order of presentation of the rules, on the specificity principle or on explicitly given information about the strengths of the rules. However, there are a few approaches which contain rules that can defeat defaults but that cannot be used to derive something. The best known are the undercutting rules in argumentation theories. If permissions are used to indicate exceptions, then they seem to play the same role as undercutters.

So this analogy with nonmonotonic logics is a structural motivation for the existence of permissive norms distinct from imperative norms.

Unluckily, there is still another counterargument. For Opalek and Wolenski [25], also permissions as exceptions seem to be superfluous: “the concept of strong permission does not perform any essential part in the analysis of the normative discourse”, in fact, “the introduction of the sphere of strong permission into the universe of normative regulation results in a strong permission becoming something of the kind of strong indifference”, p.181. This position has been also recently supported by Dignum and Royakkers [30].

Alchourron and Bulygin [2] argue that this position is wrong: “if p is strongly permitted then $\neg p$ must be strongly permitted as well. [...] This conclusion is correct only on the condition that there is no other norm concerning p ”, p.361.

It is possible that “there is a norm permitting p ; p is strongly permitted, but $\neg p$ is not regulated (it belongs to the extranormative sphere)”, p.362.

For example, it is forbidden to have guns, but it is permitted for policemen to have guns. Is it permitted to policemen not to have a gun? It depends on other norms. Maybe, later an obligation that security forces have guns is added, and hence policemen are both permitted and compelled to have guns.

Once that the need for strong permission is established, then it still seems that a permission seems *always* to presuppose the existence of a corresponding prohibition. A weak permission $P_w(a)$ amounts to the mere negation of a corresponding prohibition to a $\neg O(\neg a)$, so it does not have an autonomous normative character. A strong permission $P_s(a)$ corresponds explicitly to a permissive norm stating that a is permitted, and it constitutes an exception to a prohibition $O(\neg a)$. But given such a definition, a permission seems to lack a purpose if there is no corresponding prohibition.

2.4 Permissions in hierarchical legal systems

For Bulygin there is more to permissive norms than this: “the role played by permissive norms is not exhausted by derogation of former prohibition: an act of permitting an action which has not been hitherto prohibited is not at all pointless as has been suggested by those who deny the importance of permissive norms” [13, p.213].

Bulygin explains why permissive norms are not superflu-

ous even if there is no corresponding prohibition. What is lacking in the previous discussion is the idea that a normative system is not composed of only a single authority which enacts norms. Rather, a normative system is composed of many authorities which are linked by hierarchical relations. Moreover, a normative system has a dynamic character: norms are added to the system one after the other and this operation is performed by different authorities at different levels of the hierarchy.

Bulygin [13] notes that it is only in this interpretation of a normative system that permissive norms show their full relevance: “only in a dynamic perspective of a hierarchically structured normative system (with a plurality of norm authorities belonging to different levels) that changes in the source of time as a result of different normative acts carried out by norm authorities where the concept of a permissive norm becomes really fruitful”, p.216.

Also Alchourron and Makinson [3] support this view: “when we consider the regulations in legal or administrative code, we can often discern some kind of hierarchy among them. Some are regarded as more basic or fundamental than others”, p.125.

2.5 Meta-norms

In order to study a notion of permission in deontic logic which respects these requirements we introduce in this paper a legal system structured in a hierarchy of norm authorities that can enact both obligations and permissions. Once we have a plurality of authorities we need to cope with conflicts: our solution is the use of *meta-norms*. The meta-norms of the system ascribe to each level of authority an area of competence (a set of propositions they can permit or forbid) and prescribe that the system must respect normative principles like “lex superior derogat inferiori” (“norms have the function of preventing - inhibit, preclude - the creation of imperative norms by subordinated sources of law”, [16, p.29]), “lex posterior derogat priori” (the function of abrogating preexisting imperative norms or to derogate to them) and “lex specialis derogat generali” (the function to issue exceptions to coexisting and adjacent imperative norms). I.e., the norms of an authority have the priority over the (conflicting) norms stated by lower-level authorities and the norms it previously stated (also those which have a more general character); where “norm” means either an obligation or a permission.

Such principles are not logical principles, but rather meta-norms on the *validity* of norms, so we keep them distinct from the core of the logic: in our framework, different legal systems can be characterized by different meta-norms.

We adopt the following simple notion of validity: “A norm is *formally valid* in a normative order On when it is produced in conformity with the meta-norms which in that order regulate the creation of norms. [...] A norm is *materially valid* in an ordering On when its content is not incompatible with the other norms in On which are hierarchically superior” [26], p.197. Where the incompatibility is explained by the meta-norms of the system.

These severe requirements can be satisfied only in a framework which is able to deal with conflicts among norms, conditional norms with more or less specific antecedents and with a priority relation over norms. For these reasons, our proposal is based on an extension of input/output logic [20] with permissions as exceptions and hierarchies.

3. INPUT/OUTPUT LOGIC

Input/output logic takes its origin in the study of conditional norms. These may express desired features of a situation, obligations under some legal, moral or practical code, goals, contingency plans, advice, etc.

To be more accurate, input/output logic has its source in a tension between the philosophy of norms and formal work of deontic logicians. Philosophically, it is widely accepted that a distinction may be drawn between norms on the one hand, and declarative statements on the other. Declarative statements may bear truth-values, in other words are capable of being true or false; but norms are items of another kind. They may be respected (or not), and may also be assessed from the standpoint of other norms, for example when a legal norm is judged from a moral point of view (or vice versa). But it makes no sense to describe norms as true or false.

Input/output logic does not treat conditional norms as bearing truth-values. They are not embedded in compound formulae using truth-functional connectives. To avoid all confusion, they are not even treated as formulae, but simply as ordered pairs (a, x) of purely boolean (or eventually first-order) formulae. Technically, a normative code is seen as a set G of conditional norms, i.e., a set of such ordered pairs (a, x) . For each such pair, the body a is thought of as an input, representing some condition or situation, and the head x is thought of as an output, representing what the norm tells us to be desirable, obligatory or whatever in that situation. The task of logic is seen as a modest one. It is not to create or determine a distinguished set of norms, but rather to prepare information before it goes in as input to such a set G , to unpack output as it emerges and, if needed, coordinate the two in certain ways. A set G of conditional norms is thus seen as a transformation device, and the task of logic is to act as its ‘secretarial assistant’.

We start with obligations. In this paper we only consider the proof theory of input/output logic [20], not its semantics.

DEFINITION 1 (OBLIGATIONS). *Let L be a base logic with \top a tautology, and let G be a set of ordered pairs of L (called the generators). A generator (a, x) is read as ‘if input a then output obligatory x ’. An input/output logic out is a closure operation on $G \cup \{(\top, \top)\}$ under replacement of logical equivalents, the rules *SI*, *WO* and *AND*, and a subset of *OR*, *CT* and *ID*.*

$$\begin{array}{lll} \text{SI} & \frac{(a, x)}{(a \wedge b, x)} & \text{WO} & \frac{(a, x)}{(a, x \vee y)} & \text{AND} & \frac{(a, x), (a, y)}{(a, x \wedge y)} \\ \text{OR} & \frac{(a, x), (b, x)}{(a \vee b, x)} & \text{CT} & \frac{(a, x), (a \wedge x, y)}{(a, y)} & \text{ID} & \frac{}{(\overline{a}, a)} \end{array}$$

EXAMPLE 1. *Given $G = \{(a, x), (a, y), (x, z)\}$ the output of G contains $(a \wedge b, x)$, $(a \wedge x, z)$, $(a, x \vee y)$, $(a, a \vee x)$, and $(a, x \wedge y)$ using rules *SI*, *WO* and *AND*. Using also the *CT* rule, the output contains (a, z) .*

3.1 Permissions

Permissions are more ambiguous than obligations, and different notions have been defined.

Makinson and van der Torre [22] distinguish three notions of permission. First, *negperm* is the negation of an obligation, it corresponds to what is called weak permission. Second, *statperm* guides the citizen in the deontic assessment of specific actions, and behaves like a weakened obligation: given what is obligatory and what is strongly permitted the actual permissions of an agent are computed.

They are called weakened obligations since $\text{statperm}(P, G) \subseteq \text{out}(P \cup G)$, see [22] for details. Third, *dynperm* guides the legislator by describing the limits on what may be prohibited without violating static permissions, which is called prohibition immunity: “on the other hand, dynamic permission corresponds to the needs of the legislator, who needs to anticipate the effect of adding a prohibition to an existing corpus of norms. If prohibiting x in condition a would commit us to forbid something that has been positively permitted in a certain realizable situation, then adding the prohibition is inadmissible under pain of a certain kind of incoherence, and the pair (a, x) is to that extent immune from prohibition. For this reason, dynamic permission could also be called prohibition immunity” [22].

DEFINITION 2 (PERMISSIONS). *Let G and P be two sets of generators, where P stands for permissive norms, and let out be an input/output logic.*

- $(a, x) \in \text{negperm}(G)$ iff $(a, \neg x) \notin \text{out}(G)$;
- $(a, x) \in \text{statperm}(P, G)$ iff $(a, x) \in \text{out}(G \cup Q)$ for some singleton or empty $Q \subseteq P$;
- $(a, x) \in \text{dynperm}(P, G)$ iff $(c, \neg z) \in \text{out}(G \cup \{(a, \neg x)\})$ for some pair $(c, z) \in \text{statperm}(P, G)$ with c consistent.

EXAMPLE 2. *It is obligatory to make homework, but if one does homework he is permitted to watch the television $G = \{(\top, h)\}$, $P = \{(h, w)\}$. Then $(\top, h) \in \text{negperm}(G)$, since what is obligatory is permitted and $(a, b) \in \text{negperm}(G)$ since given a there is no restriction about b . Moreover, $(h, w) \in \text{statperm}(P, G)$ since this is explicitly permitted and $(a, w) \in \text{dynperm}(P, G)$: $(a \wedge h, \neg w) \in \text{out}(G \cup \{(a, \neg w)\})$ for some pair $(a \wedge h, w) \in \text{statperm}(P, G)$.*

3.2 Constraints

The main problem of reasoning with obligations and permissions is the question how to deal with violations and obligations resulting from violations, known as contrary-to-duty reasoning. It has been discussed in the context of the notorious contrary-to-duty paradoxes such as Chisholm’s and Forrester’s paradox. It has led to the use of constraints in input/output logics [21].

The strategy is to adapt a technique that is well known in the logic of belief change - cut back the set of norms to just below the threshold of making the current situation contrary-to-duty. In effect, input/output logic carries out a contraction on the set G of generators.

In input/output logics under constraints, a set of generators and an input does not have a set of propositions as output, but a set of set of propositions. We can infer a set of propositions by for example taking the join (credulous) or meet (sceptical), or something more complicated. Besides, we can adopt an output constraint (the output has to be consistent) or an input/output constraint (the output has to be consistent with the input). In this paper we only consider the input/output constraints.

DEFINITION 3 (CONSTRAINTS). *Let G be a set of generators and out be an input/output logic. Moreover, we write $x \in \text{out}(G, a)$ iff $(a, x) \in \text{out}(G)$. We define:*

- $\text{maxfamily}(G, a)$ is the set of \subseteq -maximal subsets G' of G such that $\text{out}(G', a) \cup \{a\}$ is consistent.

- $outfamily(G, a)$ is the output under the elements of $maxfamily$, i.e., $\{out(G', a) \mid G' \in maxfamily(G, a)\}$.
- $(a, x) \in out_{\cup}(G)$ iff $x \in \cup outfamily(G, a)$
 $(a, x) \in out_{\cap}(G)$ iff $x \in \cap outfamily(G, a)$

In case of contrary to duty obligations, the input represents something which is inalterably true, and an agent has to ask himself which obligations (output) this input gives rise to: even if the input should have not come true, an agent has to “make the best out of the sad circumstances” [18].

Makinson and van der Torre [21] consider the following example.

EXAMPLE 3. *Multiple level of violation may be analyzed. For example, put $G = \{(\top, \neg a), (a, x), (a \wedge \neg x, y)\}$ where a is read as ‘you break your promise’, x as ‘you apologize’ and y as ‘you are ashamed’. Consider the input $a \wedge \neg x$. Then $out(G, a \wedge \neg x) = Cn(\neg a, x, y)$, which is consistent. On the other hand, $out(G, a \wedge \neg x)$ is inconsistent with input $a \wedge \neg x$, so that $maxfamily(G, a \wedge \neg x) = \{(a \wedge \neg x, y)\}$ and $outfamily(G, a \wedge \neg x) = \{Cn(y)\}$.*

Permissions under constraints can be formalized by replacing in Definition 2 each occurrence of out by out_{\cup} or out_{\cap} . In this paper we make two observations. First, $negperm$ and $statperm$ still behave the same, and in particular $statperm$ behaves like a weakened obligation and guides the citizen in the deontic assessment of specific actions. However, $dynperm$ no longer formalizes prohibition immunity, because adding a conflicting obligation does not necessarily create an incoherence. For example, Sartre argues that it is coherent that it is forbidden to kill $((\top, \neg k) \in G)$, but a soldier is obliged to kill in wartime $((w, k) \in G)$. We have $(w, k), (w, \neg k) \in out_{\cup}(G)$ and $(w, k), (w, \neg k) \notin out_{\cap}(G)$.

The process of adding norms to a normative system must be seen in its diachronic character: norms are added to derogate the previous ones. Rather than checking the consistency of a new norm the question is which norms are still materially valid after a new one has been added.

As we discuss in Section 5.5 the process of adding new norms to a system must take into account the meta-norms concerning the material validity of norms when a new norm, either an obligation or a permission, derogates or abolishes previous ones.

Second, the three notions of permission do not cover permissions as exceptions, while, as we have seen in Section 2.4 this is the principal role of permissive norms in legal systems.

Most exceptions in the criminal code can be understood as such permissions, e.g., consider “it is forbidden to kill $((\top, \neg k) \in G)$, but it is permitted to kill in self-defense $((s, k) \in P)$, unless a policeman is killed $((s \wedge p, \neg k) \in G)$ ”. In input/output logic with constraints, these norms still imply the prohibition to kill in case of self-defense $((s, \neg k) \in out_{\cup/\cap}(G))$, because $maxfamily$ and $outfamily$ do not take permissions into account.

In the following section we consider the formalization of these permissions. To formalize them, we have to introduce a way for permissions to block or override obligations.

Again it is the dynamics of a normative system that highlights the importance of permissions-as-exceptions. In fact, it is possible that a permission is enacted in the same norm as the obligation it is an exception to, but it is also possible

that further exceptions to this obligation are introduced afterwards: it is possible to modify the normative system by adding new permissions after the introduction of the obligations they are exceptions to. Finally, it is possible to introduce obligations which are exceptions to some previously introduced permission.

4. PRIORITIES AND HIERARCHIES

Without permissions, a conflict can be defined in input/output logic as a case in which $outfamily$ contains more than one element. Conflict resolution has been studied in defeasible deontic logic. It can be formalized by an ordering on the powerset of generators, such that a $preffamily$ selects from $maxfamily$ only the preferred elements. An extension needed here is the distinction between generator pointers and the generators themselves, because the same generator may occur several times in the ordering. In fact, the same generator can be the object of norms enacted by different authorities: however, all these instances of the generator may have different priorities. So, we consider each norm, i.e., each instance of a generator, as a different generator pointer.

As usual we assume that the ordering is at least a partial pre-order, i.e., antisymmetric and transitive, and that it contains the subset-ordering.

With permissions, there are many ways in which the system can be extended. In this paper we use the following definition. It is inspired by $statperm$, since permissions are treated as weak obligations.

DEFINITION 4 (PERMISSIONS AS EXCEPTIONS). *Let G and P be disjoint sets of generators pointers, V a function that associates with every generator pointer a generator, and \leq a partial pre-order on the powerset of $G \cup P$ that contains the subset-ordering. We read $A \leq B$ as B is preferred to A .*

- $maxfamily(G, P, V, a)$ is the set \subseteq -maximal $G' \cup P'$ such that $G' \subseteq G$, $P' \subseteq P$ and $out(V(G') \cup V(P'), a) \cup \{a\}$ is consistent for every singleton or empty $Q \subseteq P'$.
- $preffamily(G, P, V, \leq, a)$ is the set of \leq maximal elements of $maxfamily(G, P, V, a)$.
- $outfamily(G, P, V, \leq, a)$ is the related output, i.e., $\{out(V(G'), a) \mid G' \cup P' \in preffamily(G, P, V, \leq, a), G' \subseteq G, P' \subseteq P\}$.
- $statpermfamily(G, P, V, \leq, a)$ is defined analogously, $\{out(V(G' \cup Q), a) \mid G' \cup P' \in preffamily(G, P, V, \leq, a), G' \subseteq G, Q \subseteq P' \subseteq P, Q \text{ is a singleton or empty}\}$.
- $out_{\cup/\cap}(G, P, V, \leq)$ are analogous as in Definition 3.

The following example illustrates permissions as exceptions, but also obligations as exceptions to permissions.

EXAMPLE 4. *Let $G = \{a = (\top, \neg k), b = (s \wedge p, \neg k)\}$, $P = \{c = (s, k)\}$, $\{a, c\} < \{a, b\} < \{b, c\}$, where by $A < B$ we mean as usual $A \leq B$ and $B \not\leq A$.*

It is forbidden to kill, but it is permitted to kill in case of self-defence, unless a policeman is killed.

$maxfamily(G, P, V, \leq, s) = \{\{a, b\}, \{b, c\}\}$,

$preffamily(G, P, V, \leq, s) = \{\{b, c\}\}$,

$outfamily(G, P, V, \leq, s) = \{Cn(\{\top\})\}$,

$statpermfamily(G, P, V, \leq, s) = \{Cn(\{k\})\}$

The *maxfamily* includes the sets of applicable compatible generators together with all non applicable ones: e.g., the output of $\{a, c\}$ in the context s is not consistent. Even if b could conflict with c , it is not applicable in a situation s (while b is), hence $\{b, c\}$ is consistent. Finally $\{a\}$ is not in *maxfamily* since it is not maximal, we can add the non applicable rule b . Then *preffamily* is the preferred set $\{b, c\}$ according to the ordering on set of rules above. The set *outfamily* is composed by the consequences of applying the generators whose pointer is in G which are included in $\{b, c\}$ which are applicable in s : b is the only obligation, but it is not applicable. The set *statpermfamily* is the result of the application of obligations in G together with a permission at a time from P , in this case c .

Conflicts between obligations and permissions are traditionally studied using hierarchies [3], which we formalize as an ordering on the generator pointers, in which each cluster represents an authority.

Alchourron and Makinson [3] define a hierarchy of regulation in this way: “a *hierarchy of regulations* to be a pair (A, \leq) where A is a non-empty set of propositions, called a *code*, and \leq is a partial ordering of A ”, [3, p.126]. But this definition is not sufficient since we must compare sets of norms: “it will sometimes happen that no one regulation by itself suffices to yield a verdict, but two or more taken together do. Thus the judge, and we, need to consider *sets* of regulations”. “[The judge] need[s] to compare, whenever possible, one set of regulations with another. In other words, given a relation \leq that partially orders A , we need to envisage ways which \leq induces some kind of ordering of 2^A ”, [3, p.127]. Our definition above considers a relation \leq on set of rules and not a relation on rules.

Thus, given a relation on rules, how can it be lifted to a relation on sets of rules? As Alchourron and Makinson [3] notice, “there is no *a priori* way of determining what the most suitable definitions will be”. For example, they propose the following one: “if (A, \leq) is a hierarchy of regulations and $B, C \subseteq A$ we shall [...] write $C \leq B$, iff for every $b \in B$ there is a $c \in C$ with $c \leq b$ ”, p.127 (where C is preferred over B). But such a definition has the consequence that if there is a $c \in C$ such that for all $b \in B$ $b \leq c$ then $B \leq C$. And in particular if $p \leq q \leq r$ then $\{p, r\} \leq \{q, r\}$ and $\{q, r\} \leq \{p, r\}$, even if we would expect that $\{q, r\}$ would be preferable, since $p \leq q$.

So we adopt a different definition of lifting, inspired by Brass [10]: “ $E_1 \leq E_2 \iff$ for every $\delta_2 \in E_2 \setminus E_1$ there is a $\delta_1 \in E_1 \setminus E_2$ with $\delta_1 \preceq \delta_2$.”² Hence, we give the following way in which priorities and a hierarchy can be related.

DEFINITION 5 (HIERARCHY OF NORMS). A *hierarchy* is a partial pre-order \preceq on generator pointers. A *priority ordering* on set of rules \leq respects \preceq when $B \leq C$ if for every $b \in B \setminus C$ there is a $c \in C \setminus B$ with $b \preceq c$. We write $a \prec b$ for $a \preceq b$ and $b \not\preceq a$.

A *hierarchical normative system* is a legal system where norms are arranged in a hierarchy of norms.

EXAMPLE 5. Given the relation on set of rules used in Example 4 $\{a, c\} < \{a, b\} < \{b, c\}$ which is the ordering on

²Where with respect to [10] we reverse the meaning of the \leq relation: for Brass $B \leq C$ means that B is preferred over C .

rules that it respects? $a \prec c \prec b$, in fact, a is a general obligation which is derogated by c in a situation of self-defence. Obligation b is an exception to permission c , so it must have precedence over c . The restriction above is respected: e.g., $\{a, b\} \setminus \{a, c\} = \{b\}$, $\{a, c\} \setminus \{a, b\} = \{c\}$ and $a \preceq c$.

Even if in this paper we do not examine the logical properties of this definition of priority, we notice that our definition respects the *principle I* of Brewka and Eiter [11]’s work on prioritized default theories: “let B_1 and B_2 be two extensions of a prioritized default theory Δ generated by the defaults $R \cup \{d_1\}$ and $R \cup \{d_2\}$ where $d_1, d_2 \notin R$, respectively. If d_1 is preferred over d_2 , then B_2 is not a preferred extension.”

Also Brewka and Eiter [11]’s *principle II* that adding a rule which is not applicable in a preferred belief set can never render this belief set non-preferred is respected; in fact, since the extensions of generator pointers in *maxfamily* are maximal sets of generators, all generators which are not applicable are included in the set.

5. STATIC AND DYNAMIC NORMS

We now consider the distinction between static and dynamic norms.

5.1 Static norms

The static interpretation makes sense if the hierarchical normative system contains a single authority: then norms should not be weakly redundant.³

DEFINITION 6 (STATIC NORMS). $g \in G \cup P$ is *weakly redundant* iff $\forall a \in L : \text{outfamily}(G, P, V, \leq, a) = \text{outfamily}(G \setminus \{g\}, P \setminus \{g\}, V, \leq, a)$.

In a static normative system permissions are meaningful only if they *derogate* some other obligations, in part or *in toto*: in the definition to decide whether g is weakly redundant we compute the set of what is obligatory in the normative system, or *outfamily*; adding a permission which does not affect the material validity of some obligation does not affect *outfamily*.

In the same way as permissions, obligations can have a role of derogating some permission (as in Example 4) or even another obligation, as the following example shows:

EXAMPLE 6. $G = \{a = (\top, \neg f), b = (p, f)\}, P = \emptyset, \{a\} < \{b\}$,
No one should have firearms, policemen should have one.
 $\text{maxfamily}(G, P, V, \leq, p) = \{\{a\}, \{b\}\}$,
 $\text{preffamily}(G, P, V, \leq, p) = \{\{b\}\}$,
 $\text{outfamily}(G, P, V, \leq, p) = \{Cn(\{f\})\}$,
 $\text{statpermfamily}(G, P, V, \leq, p) = \{Cn(\{f\})\}$

In contrast, in the following example we consider a permission which is weakly redundant:

EXAMPLE 7. $G = \{a = (\top, p)\}, P = \{b = (q, p)\}, \{a\} \leq \{b\}$,

³Alternative notions can be based on $\text{out}_{\cup/\cap}$. Due to space limitations in this and the following definition we skip some details about \leq and V , since they have to shrink or grow in the obvious way with the removal or addition of norms. See again Brewka and Eiter [11] for a definition of agreement between two priority relations.

One should pay taxes and if he is rich he is permitted to pay taxes.

In context q , we have:

$\text{maxfamily}(G, P, V, \leq, q) = \{\{a, b\}\}$,

$\text{preffamily}(G, P, V, \leq, q) = \{\{a, b\}\}$,

$\text{outfamily}(G, P, V, \leq, q) = \{Cn(\{p\})\}$,

$\text{statpermfamily}(G, P, V, \leq, q) = \{Cn(\{p\})\}$

but b is weakly redundant since the set of obligations does not change if b is removed:

$\text{outfamily}(G \setminus \{b\}, P \setminus \{b\}, V, \leq, q) = \{Cn(\{p\})\}$,

5.2 Dynamic norms

The dynamic interpretation is relevant when the hierarchy contains multiple authorities: then, an authority may introduce a statically redundant norm to block the possibility that lower level authorities introduce conflicting and materially valid norms. It is strongly redundant to oblige or permit something which is permitted or forbidden at a higher level, and to permit something which cannot be obligated by the lower levels, for example because they have no competence on it. In such a setting, we may say that the lower and higher levels of authorities play a game against each other.

First we define a simple notion of competence, as a set of propositions associated to each authority.

DEFINITION 7 (COMPETENCE AND FORMAL VALIDITY).

The competence of an agent is a function C which associates a set of propositions in L to each authority in A . An authority a in the set of authorities A of the hierarchical legal system has competence over a norm (x, y) , if $y \in C(a)$.

If an authority has no competence over a norm (x, y) then the norm cannot be formally valid and thus it cannot be added to the set of norms $G \cup P$ of the normative system.

DEFINITION 8 (DYNAMIC NORMS). $g \in G \cup P$ is strongly redundant iff $\forall a \in L, \forall G', P'$ sets of formally valid norms $\forall g' \in G' \cup P' : g' \prec g$ then $\text{outfamily}(G \cup G', P \cup P', V, \leq, a) = \text{outfamily}(G \setminus \{g\} \cup G', P \setminus \{g\} \cup P', V, \leq, a)$.

DEFINITION 9 (MATERIAL VALIDITY). A norm which is not weakly or strongly redundant is said to be materially valid in the system.

The following example illustrates the distinction between static and dynamic norms.

EXAMPLE 8 (CONTINUED FROM EXAMPLE 4). If there is a single authority, and we only consider the rules $\{a, b\}$, then b is strongly redundant. If there are multiple authorities a_1 and a_2 , $a_1 \preceq a_2$ such that $a \prec b$, then b is again redundant. However, if we consider all three rules $\{a, b, c\}$ with $a \prec c \prec b$, then none of the rules is redundant. Hence, with multiple authorities and the rules $\{a, b\}$, b is not strongly redundant if there is a lower level authority who has the competence to enact c .

This notion is strictly related to the dynamics of the normative system: what happens when a new norm is added? And in particular: when can a norm, and specifically a permission, be meaningfully added to the system?

The latter definition of strong redundancy supports Bulygin's argument in favor of permissions which do not presuppose explicitly existing obligations: in fact, the new permission prevents lower level authorities to forbid the permitted

behavior *in toto* or under some circumstances; such a permission creates a (would be) prohibition immunity (with respect to lower level authorities). Analogously, an obligation creates a "permission immunity" with respect to lower level authorities.

Even if in this work permissions maintain an independent character with respect to obligations (in contrast with the so called "imperativist" view of legal systems), it must be noticed an asymmetry between obligations and permissions: in fact, an obligation is meaningful even if it does not derogate or abolish a permission nor there is no lower-level competent authority: the new legal system after the addition of the obligation can modify the citizen's behavior. The definitions of redundancy are given in terms of what is obligatory and not of what is permitted. In a sense, however, also meaningful obligations always derogate some permission: either a strong one or a weak permission; in the latter case, the weak permission can be something which is indifferent (not obligated nor prohibited) or which has been prescribed by another obligation: thus, the new obligation derogates to this older one.

EXAMPLE 9. Consider the sets of obligations and permissions $G = \{a = (\top, \neg p), b = (q, r), c = (s, p), d = (s, \neg t)\}$ and $P = \{e = (\top, t)\}$ with $a \prec b, a \prec c, e \prec d$,

The norm b derogates to the negative permission $(\top, \neg r)$, due to the fact that r is indifferent without b . The norm c derogates to permission $(\top, \neg p)$, a negative permission following the corresponding obligation a . Finally, d derogates the strong permission e . We have:

$\text{maxfamily}(G, P, V, \leq, q \wedge s) = \{\{a, b, e\}, \{a, b, d\}, \{b, c, d\}, \{b, c, e\}\}$,

$\text{preffamily}(G, P, V, \leq, q \wedge s) = \{\{b, c, d\}\}$,

$\text{outfamily}(G, P, V, \leq, q \wedge s) = \{Cn(\{r, p, \neg t\})\}$,

In this section we raised several issues, among them the hierarchical relations on norms, the relation of the notion of competence with norms and the relation between permissions and obligations. In the following sections we will discuss in more depth these arguments.

5.3 Validity and meta-norms

First of all the \preceq hierarchy of norms is not necessarily primitive but it can be generated according to some principles.

These principles play the roles of *meta-norms* which, according to Mazzaresse [23], "establish which norms do constitute a given legal order", i.e., in our terminology, meta-norms establish which norms are materially valid.⁴ Moreover, as in [23], we do not suppose that validity requires no conflict among norms. Rather the meta-norms which define validity are the conceptual tool which regulates how a legal order deals with conflicts.

DEFINITION 10 (META-NORM: LEX SUPERIOR).

Given an ordering \preceq_A on the set of authorities A , two authorities $a_1, a_2 \in A$, two norms $n_1, n_2 \in G \cup P$ $n_1 \preceq n_2$ if $a_1 \preceq_A a_2$. This expresses the principle "lex superior derogat legi inferiori".

⁴In contrast with the notion of systemic validity of [24], we keep separate the problems of formal and material validity since in this paper we do not address the first problem, apart from the limited treatment of competence.

Or we can take into account the time of creation of a norm.

DEFINITION 11 (META-NORM: LEX POSTERIOR).

Given a function T which associates each generator pointer to a timestamp i in the set of natural numbers (which represents the time at which the norm has been issued by an authority), then, given $n_1, n_2 \in G \cup P$, $n_1 \preceq n_2$ if $T(n_1) \leq T(n_2)$. In this way, we express the principle “lex posterior derogat legi priori”.

Finally, the two principles can be combined so that first superior norms are considered and then, if two norms are issued by authorities of the same level, the time of creation is considered:

DEFINITION 12 (LEX POSTERIOR AND SUPERIOR).

Given $a_1, a_2 \in A$, two norms $n_1, n_2 \in G \cup P$, $n_1 \preceq n_2$ if $a_1 \preceq_A a_2$ and if $a_2 \preceq_A a_1$ then $T(n_1) \leq T(n_2)$.

If the ordering \prec is a total order (there are no authorities with the same competence at the same level and the function T is injective), then the *preffamily* is a singleton.

5.4 Validity and competence

Further comments are needed about the relation between the notion of competence and formal validity. Similarly to Mazzarese [23], we do not have a reductionist view like the one of Ross [29] who supports the idea that “any norm of competence may be transcribed as a norm of conduct” p.120.

Note that in our model, a non-redundant permission which does not derogate a preexisting obligation does not correspond to a limitation of the competence of a lower level authority. In fact, competence is defined as a set of propositions, while a higher level permission is expressed as a conditional rule. So a permission can derogate an eventual norm only in part and not *in toto*, thus non hindering the competence of the lower level authority in those cases where the condition of the permission is false. We analyze this possibility in the following example:

EXAMPLE 10. Consider a hierarchical normative system where the “lex superior” meta-norm holds. An authority a_2 has the competence about d ($d \in C(a_2)$), e.g. it can permit or forbid to download programs from the web. However, the higher rank authority a_1 , $a_2 \prec_A a_1$ which also has the competence d , decides to permit downloading of antivirus programs, $(a, d) \in P$. This permission does not hinder a_2 ’s competence d in that a_2 can still forbid the download of different types of programs, e.g. crackers $(c, \neg d) \in G$, but also more general ones, e.g. every program $(\top, \neg d) \in G$. These norms are both formally valid, because $d \in C(a_2)$ is still true, and also materially valid. For example, we have $\neg d \in \text{outfamily}(\{(c, \neg d)\}, \{(a, d)\}, V, \leq, c)$.

However, the prohibition $(\top, \neg d)$ does not cancel the higher level permission in case the condition a , the program is an antivirus, is true (even if a implies \top):

$\neg d \notin \text{outfamily}(\{(\top, \neg d)\}, \{(a, d)\}, V, \leq, a)$
 $d \in \text{statpermfamily}(\{(\top, \neg d)\}, \{(a, d)\}, V, \leq, a)$

We believe that a further argument in favor of this position is the principle of subsidiarity, accepted, e.g., by the EU regulations: all the norms should be enacted by the lowest level of authority which is competent about them. The existence of such principle suggests that a higher level norm

does not limit the competence of lower level authorities, but rather their possibility to enact materially valid norms.

This principle can be defined in our framework in the following way:

DEFINITION 13 (SUBSIDIARITY). A normative system respects the principle of subsidiarity if for all norms $(a, b) \in G \cup P$ enacted by authority a_1 there is no authority a_2 such that $a_2 \preceq a_1$ and $b \in C(a_2)$.

Finally, we return on the problem that permissions seem to presuppose obligations.

5.5 Rational creation of norms

As in Mazzarese [23] we do not identify derogation merely with the absence or the (external) negation of a corresponding prohibition. Rather it is a phenomenon regulated by meta-norms and has a diachronic character: the withdrawing of the material validity of a norm presupposes that the norm was materially valid and that “it becomes no longer valid after the derogation, because of the derogation” [23]. Moreover also the thetic character of a derogation is respected, i.e., it “amounts to a cancellation, the repealing of the systemic validity of a legal norm”. Hence we believe not to be affected by Mazzarese [23]’s criticism of deontic logic approach to validity.

In some definitions of strong permissions, permissive norms are seen as implying or presupposing a corresponding obligation which the permission is an exception to. For example Ross [29] and Dignum and Royakkers [30] (see Section 6). As Bulygin [13] has shown, this presupposed obligation does not necessarily is an already existing one, but rather an eventual one. So those models where permissions presuppose obligations in their definition do not take Bulygin’s argument into account.

In our model, we explained the idea of a presupposed obligation in terms of the rationality of the norm creation process: the non redundancy of norms. In this way, the requirement that permissions are exceptions to obligations becomes a *pragmatic* principle: it is not meaningful to add a permission which is not an exception and which cannot be an exception to an eventual obligation. If a permission is not redundant then it presupposes an obligation. But a legislator can anyway add redundant norms, since there is no limitation to this possibility at the syntactic or semantic level.

As Grice [15] has shown for linguistic principles, such pragmatic constraints can be violated to express a meaning anyway. For example, a lower rank authority can issue a permission to smoke dope which it knows it is not materially valid (i.e., it is blocked by an higher level prohibition to use dope) only to show to its voters its willingness to deal with that issue. Or a higher rank authority can permit something which is already permitted by all lower level authorities to emphasize the importance of such permission.

Second the pragmatic principles we propose have a more general character: they do not apply only to permissions, but to obligations as well. There can be in fact redundant obligations whose creation does not change the duties and permissions of the citizens; in fact, they are cancelled by an higher level permission or they are already implied by some other obligations. Again, an authority can forbid dope even if it is permitted by more important norms just to show its disagreement with the current normative system.

6. RELATED WORK

Alchourron and Makinson [3] first addressed in a formal way the notion of derogation: “the [legislative] body may decide to reject y , with the intention of thereby rejecting implicitly whatever in A implies y , retaining the remainder. This we shall call derogation”, p.127. Moreover “in the special case where Y is a subset D of A , to derogate it, in the sense we have specified is to do much more than merely rescind or abrogate it. For when we abrogate a subset D of A , we merely *drop* it from the code, leaving $A - D$ intact even if it implies some of the regulations in D ”, p.130. This model subsequently lead to a solution of the belief revision problem by Alchourron, Gärdenfors and Makinson [1].

In this paper, we are interested in a complementary problem. We are not interested in computing the remainder of a set of propositions given a norm to be derogated. Rather, we focus on the point of view of the legislator who can change the normative system by adding permissions and obligations. Given a new permission or obligation we deal with the problem of computing what is now permitted or obligatory.

Royakkers and Dignum [30] present a formalization of the distinction between weak and strong permission. The focus of the work is the fact that a permission which is enacted, i.e., a strong permission, implies that it is strongly permitted also the negation of what is permitted. The reason to this claim is that “a regulation concerning a permission is always an exception of an obligation or a prohibition [...]. Otherwise the permission would be superfluous, because of the sealing principle ‘whatever is not forbidden is permitted’.” Hence a strong permission “implies a choice for the norm subjects to perform that act or not, without a liability to sanction.”

However, this position, which is similar to the one of Opalek and Wolenski [25], has been challenged (see Section 2.3). Moreover, in this paper, we propose a different view where the relation of permissive norms with obligations is treated more flexibly at a pragmatic level; so we have not to modify the syntax or semantic of the logic to model the fact that permissions “presuppose” in some sense obligations.

The authors provide a logic of enactments in order to distinguish (unconditional) permissions which are implicit in the norms and those explicitly enacted by some authority. Since enacted norms are allowed to be inconsistent while the sets of norms which are applicable in a given situation and which the lawyers use to take a decision must be consistent, the logic of enactment is based on Fagin and Halpern [14]’s local reasoning semantics. However, Dignum and Royakkers [30] does not address the issue of determining which sets of applicable norms are preferred in a given situation, as instead we do in our proposal.

More recently Brown [12] also argues for the need of modelling strong (explicit in his terminology) permissions in deontic logic as a different normative category with respect to obligations and to express the fact that a permission is meaningful only if there is a corresponding prohibition. His proposal to model permissions as exceptions is, however, based on a monotonic logic. So to avoid a conflict between an obligation and the corresponding permission he defines obligations as implicitly conditional to the absence of a permission to do what prescribed. E.g., he models “you ought not to drive through an intersection while the traffic light is red” as “you ought not to drive through an intersection while the traffic light is red unless explicitly permitted to do so”. This treatment seems problematic to us since in case of

conditional permissions as exceptions, obligations should be defined conditional on an infinite set of permissions (since there could be an infinite number of conditions). Moreover, we provide a different type of approach where the meaningfulness of a permission is not encoded at the syntactic or semantic level but at the pragmatic one: the point of a permission depends on his non-redundancy in the normative system it is added to.

Our treatment of permissions raises the question whether permissions can be related to undercutters in argumentation theory, when obligations are related to arguments and rebutters.

In Pollock [27] rebutters are a kind of defeat of an argument which attacks a reason by supporting an opposite conclusion; undercutters are arguments that attack the fact that another reason supports its conclusion: given an argument $P \rightarrow Q$, an undercutter is a reason for $\neg(P \rightarrow Q)$.

However, when we consider conditional concerning norms, as input/output logic, we should not speak of their truth. So Pollock [27]’s definition is not adequate for obligations and permissions.

More recently, Prakken [28] proposes a treatment of undercutters in terms of premises which contains weak negation (\sim) in conditionals. An argument like $P \wedge \sim R \Rightarrow Q$ can be undercut by an argument supporting Q .

Verheij [31] criticizes this approach since he argues that it limits the possibility to add further undercutting arguments once the conditional has been created. Instead, law making has a sequential character and new permissions can be added after the obligation they are exceptions to. Hence Verheij [31] proposes a different treatment of undercutters, by introducing the unary $\times\phi$ operator with the meaning that ϕ is defeated. Undercutters are represented as conditionals which defeat other conditionals; $\phi \rightarrow \times(\psi \rightarrow \chi)$ means that ϕ is an undercutter of ψ supporting χ . The interpretation of a theory in Verheij’s DEFLOG system evaluates sentences by assigning them the values justified or defeated.

7. SUMMARY

In this paper we study permissive norms in input/output logic. Recent developments in the input/output logic framework distinguished various forms of permissions, but we show that the notion of prohibition immunity cannot be extended to the constrained setting, and that thus far the permissions as exceptions are not formalized in this framework. We introduce such a notion, and discussed some rationality constraints which can be imposed on such definitions. Thus the notion of “prohibition immunity” (and a symmetric notion of “permission immunity”) has been relativized to the different levels of authorities. Our work supports Bulygin [13] suggestion that “permissive norms are normative relevant in an indirect way”, they “are not superfluous”, i.e., “they play a characteristic role in legal discourse which could not be possibly performed by mandatory norms alone”, p.211.

In summary, in our model we formalize the following reasons why permissive norms are not superfluous:

- Permissions play the role of exceptions to obligations.
- Permissions are used for reasons of efficient communication. This may be one of the reasons behind permissions as exceptions: it is more efficient to state

general rules and exceptions than completely specified rule. Suppose an agent like to do p , but does not know whether it is forbidden or not. So he should consult the legal code, make inferences, and find out whether it is forbidden or not. However, when he is permitted, the agent does not have to worry that there is some law which prohibits p . Guibourg and Mendonca [17] call this the “*indicative function*” of norms: “the permissive norms play the function of indicating to their addressee which are the behaviors authorized by the authority who issued them”, p.280.

- Without permissions as exceptions “there would be no possibility of normative change emerging from acts of authority.” [13, p.213]. Changing a normative system would not amount to adding new obligations or permissions but only to replacing the existing norms by means of alternative ones.
- Permissions are used to block other (lower) authorities to issue norms. This is a motivation behind the dynamic interpretation of norms.

Elsewhere we have formalized obligations associated with sanctions [7, 8], extending Boella and Lesmo [6]’s proposal in a qualitative decision theory. In [9] we integrate permissions within the sanction based framework so to model also the definition of permission in terms of not being sanctionable given, e.g., in the Italian penal code.

8. REFERENCES

- [1] C. Alchourron, P. Gärdenfors, and D. Makinson. On the logic of theory change. *Journal of symbolic logic*, 50(2):510–530, 1985.
- [2] C. E. Alchourron and E. Bulygin. The expressive conception of norms. In R. Hilpinen, editor, *New studies in deontic logic*, pages 95–124. D. Reidel, Dordrecht, 1981.
- [3] C. E. Alchourron and D. Makinson. Hierarchies of regulations and their logic. In R. Hilpinen, editor, *New studies in deontic logic*, pages 125–148. D. Reidel, Dordrecht, 1981.
- [4] N. Bobbio. *Teoria della norma giuridica*. Giappichelli, Torino, 1958.
- [5] N. Bobbio. Norma. In *Enciclopedia Einaudi*, volume 9, pages 876–907, Torino, 1980. Einaudi.
- [6] G. Boella and L. Lesmo. A game theoretic approach to norms. *Cognitive Science Quarterly*, 2(3-4):492–512, 2002.
- [7] G. Boella and L. van der Torre. Attributing mental attitudes to normative systems. In *Proceedings of AAMAS’03*, 2003.
- [8] G. Boella and L. van der Torre. Rational norm creation. In *Proceedings of ICAIL’03*, Edinburgh, 2003.
- [9] G. Boella and L. van der Torre. Obligations and permissions as mental entities. In *Procs. of IJCAI Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions*, Acapulco, 2003.
- [10] S. Brass. Deduction with supernormal defaults. In *Nonmonotonic and inductive logics, LNAI 659*, pages 153–174. Springer, Berlin, 1993.
- [11] G. Brewka and T. Eiter. Prioritizing default logic. In *Intellectics and Computational Logic*, pages 27–45. Kluwer, 2000.
- [12] M. A. Brown. Conditional obligation and positive permission for agents in time. *Nordic Journal of Philosophical Logic*, 5(2):83–112, 2000.
- [13] E. Bulygin. Permissive norms and normative systems. In A. Martino and F. Socci Natali, editors, *Automated Analysis of Legal Texts*, pages 211–218. Publishing Company, Amsterdam, 1986.
- [14] R. Fagin and J. Y. Halpern. Belief, awareness and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988.
- [15] H. P. Grice. Meaning. *Philosophical Review*, 56:377–388, 1957.
- [16] R. Guastini. *Teoria e dogmatica delle fonti*. Giuffrè, Milano, 1998.
- [17] R. A. Guibourg and D. Mendonca. Permesso, garanzie, e libertà. *Analisi e Diritto*, pages 269–301, 1995.
- [18] B. Hansson. An analysis of some deontic logics. *Nôus*, 3:373–398, 1969.
- [19] D. Lewis. A problem about permission. In E. Saarinen, editor, *Essays in Honour of Jaakko Hintikka*, pages 163–175. D. Reidel, Dordrecht, 1979.
- [20] D. Makinson and L. van der Torre. Input-output logics. *J. of Philosophical Logic*, 29:383–408, 2000.
- [21] D. Makinson and L. van der Torre. Constraints for input-output logics. *Journal of Philosophical Logic*, 30(2):155–185, 2001.
- [22] D. Makinson and L. van der Torre. Permissions from an input-output perspective. *Journal of Philosophical Logic*, To appear.
- [23] T. Mazzaresse. Deontic logic as logic of legal norms: Two main sources of problems. *Ratio Juris*, 4:374–392, 1991.
- [24] T. Mazzaresse. Permesso forte e permesso debole: note a margine. *Analisi e diritto*, 2000.
- [25] K. Opalek and J. Wolenski. Normative systems, permission and deontic logic. *Ratio Juris*, 4:334–348, 1991.
- [26] F. Poggi. Obbligatorio implica permesso. *Analisi e diritto*, 2000.
- [27] J. L. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.
- [28] H. Prakken. *Logical tools for modelling legal argument*. Kluwer Academic Publisher, Dordrecht, 1997.
- [29] A. Ross. *Directives and norms*. Routledge and Kegan Paul., London, 1968.
- [30] L. Royakkers and F. Dignum. The logic of enactment. In J. Zelezniakow, editor, *Procs. of 6th International conference on AI and Law*. ACM, Melbourne, 1997.
- [31] B. Verheij. Legal decision making as dialectical theory construction with argumentations schemes. In *Procs. of ICAIL 01*, S. Louis (MI), 2001.
- [32] G. H. von Wright. *On the logic of negation*. Societas Scientiarum Fennica, Helsinki, 1959.
- [33] G. H. von Wright. *Norm and Action. A logical Inquiry*. Routledge and Kegan Paul, London, 1963.