Artif Intell Law (2008) 16:53–71
DOI 10.1007/s10506-007-9059-8

Institutions with a hierarchy of authorities in distributed dynamic environments

Guido Boella · Leendert van der Torre

Published online: 3 October 2007

Abstract A single global authority is not sufficient to regulate heterogenous agents in multiagent systems based on distributed architectures, due to idiosyncratic local situations and to the need to regulate new issues as soon as they arise. On the one hand institutions should be structured as normative systems with a hierarchy of authorities able to cope with the dynamics of local situations, but on the other hand higher authorities should be able to delimit the autonomy of lower authorities to issue valid norms. In this paper, we study the interplay of obligations and strong permissions in the context of hierarchies of authorities using *input/output logic*, because its *explicit norm base* facilitates reasoning about norm base maintenance, and it covers a variety of *conditional* obligations and permissions. We combine the logic with constraints, priorities and hierarchies of authorities. In this setting, we observe that Makinson and van der Torre's notion of prohibition immunity for permissions is no longer sufficient, and we introduce a new notion of permission as exception and a new distinction between static and dynamic norms. We show how strong permissions can dynamically change an institution 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 power of lower level authorities to change the normative system.

Keywords Normative systems · Institutions · Permissions

G. Boella (✉)
Dipartimento di Informatica, Università di Torino, Torino, Italy
e-mail: guido@di.unito.it

L. van der Torre
University of Luxembourg, Luxembourg, Luxembourg
e-mail: leendert@vandertorre.com

1 Introduction

Technological developments in multiagent systems like peer to peer systems, grid architectures, virtual organizations, and electronic institutions, pose new challenges for normative systems regulating the heterogeneous agents interacting with each other. These normative systems should be structured in a hierarchical way, because a global authority neither has complete knowledge about local situations, nor the resources to regulate new issues dynamically arising. Moreover, norms should be represented explicitly for transparency of the normative system and norm base maintenance, facilitating the introduction and removal of norms. Therefore, two branches of legal theory in general and deontic logic in particular are of crucial importance for the development of normative multiagent systems.

First, the theory of normative multiagent systems builds on the study of the use of hierarchies in normative systems. The sources of law give rise not only to norms, but also to hierarchical criteria that determine the relative importance of various norms that form part of a legal order. The identification of the material of which a legal system is composed must include a series of relations determining the relative weight of norms. The clearest example is the criterion based on the level of authority that enacted the norm. The authority hierarchy is an example of a method for filtering out the applicable set of norms from a given set of norms. Other strategies to determine the valid norms are valid in a situation are being studied as *meta-norms* by, e.g., Prakken and Sartor (1996).

Second, the theory of normative multiagent systems builds on the study of the promulgation or introduction of norms and the derogation or removal of norms and their consequences, such as the indeterminate derogation or the 'Unbestimmtheit des Normensystems' (Bulygin 1996), the problem with inconsistent systems, and the role of global authorities to provide general policies which cannot be overruled by lower authorities. Higher authorities, thus, delimit the autonomy of lower authorities of issuing valid norms. Lewis (1979) and Bulygin (1986) argue convincingly that in such cases the notion of obligation is not sufficient, as we discuss in Sect. 3, but we need a strong kind of permission too.

However, these theories developed mainly in the seventies and the eighties of the previous century cannot directly be used for normative multiagent systems, mainly because they do not make the roles the agents play within the normative system explicit, and consequently they lack detailed computational social mechanisms to maintain and reason with the norm base. Normative multiagent systems (Boella et al. 2006) seek to provide a more general framework for normative reasoning, incorporating amongst others the above multiagent issues. For example, we propose constitutive norms to define a mechanism for agents to exercise their power to update the normative system at various levels (Boella and van der Torre 2006a). Moreover, we consider Von Wright's transmission of will to formalize policies about policies in the scope of knowledge management (Boella and van der Torre 2006b). In that case, it is not sufficient that there is an obligation for the lower authority to oblige or permit the user, but the norm of the lower authority has to be enforced too. In this paper, we address the following questions:

1. How to define conditional permission as exception to conditional obligations in order to deal with the mechanism by which higher authorities can limit the norm-making of lower authorities?
2. How does the notion of strong conditional permission relate to other proposals of permission discussed in the literature?
3. How to distinguish static and dynamic norms using hierarchies of normative systems, where the former concern the derogation from (existing) obligations, while the latter regard how to block the possibility that lower authorities introduce conflicting valid norms, and how to define the competence and redundancy of norms?

We use Makinson's and van der Torre's framework of input/output logic (Makinson and van der Torre 2000), because its *explicit norm base* facilitates reasoning about norm base maintenance, and it covers a variety of *conditional* obligations and permissions. Thus far the dynamics of normative systems have been studied mainly in the framework of Alchourròn, Makinson and Gardenfors' (AGM) theory of belief revision, since Alchourron's study of retraction in legal systems was one of its constituents. However, the AGM theory seems better suited for belief revision than for norm revision, since it does not cover particularities of normative reasoning such as the distinction between norms and normative propositions, the conditional nature of norms, and the distinction between obligations and strong permissions.

Makinson and van der Torre propose input/output logic to formalize conditional norms, obligations, goals, *etc.* (Makinson and van der Torre 2000), negative, static and dynamic permissions (Makinson and van der Torre 2003), and contrary-to-duty reasoning (Makinson and van der Torre 2001). In this paper, we observe that the notion of dynamic permission no longer corresponds to so-called prohibition immunity, and that the notion of permission as exception has not been covered. We introduce priorities and to model institutions we introduce normative systems as hierarchies of authorities in input/output logic. If a normative system is structured in a hierarchical way, then norms from different authorities can conflict with each other. Moreover, authorities may change the normative status only of behaviors they are competent of.

The layout of this paper is as follows. In Sect. 2, we discuss normative multiagent systems and in Sect. 3 we summarize legal theory on permission. In Sect. 4, we present input/output logic and in Sect. 5 we introduce permissions as exceptions using conflict resolution. In Sect. 6, we make the distinction between static and dynamic norms using hierarchies of normative systems, discussing the notion of competence and of redundancy of norms.

2 Normative multiagent systems

There are numerous approaches of social norms and laws in multiagent systems (Dignum 1999; Dignum et al. 2004; Esteva et al. 2001, 2004; Lopez y Lopez et al. 2002; Shoham and Tennenholtz 1995). Jones and Carmo (2001) define a normative system as "Sets of agents whose interactions are norm-governed; the norms prescribe how the agents ideally should and should not behave. [...] Importantly, the norms allow for the possibility that actual behavior may at times deviate from the ideal, i.e., that violations of obligations, or of agents' rights, may occur." Since the agents' control over the norms is not explicit here, we use the following definition: "A normative multiagent system is a multiagent system together with normative systems in which agents can decide whether to follow the explicitly represented norms, and the normative systems specify how and in which extent the agents can modify the norms" (Boella et al. 2006). Note that this definition makes neither presumptions about the internal architecture of an agent nor of the way norms affect agent's behavior.

The question should be raised *how* the explicit norms are represented. Norms can be interpreted as a special kind of constraint, and represented depending on the domain in which they occur. However, this representation runs into the question what happens when norms are violated. Not all agents behave according to the norm, and the system has to sanction violations or reward good behavior. Thus, the normative system has to monitor the behavior of agents and enforce the sanctions. Also, when norms are represented as domain dependent constraints, the question will be raised how to represent permissive norms, and how they relate to obligations. Whereas obligations and prohibitions can be represented as constraints, this does not seem to hold for permissions. For example, how to represent the permission to access a resource under an access control system? Finally, when norms are represented as domain dependent constraints, how do norms evolve?

We therefore believe that norms should be represented as a domain independent theory. For example, deontic logic (von Wright 1951; van der Torre and Tan 1999; van der Torre 2003; Makinson and van der Torre 2000, 2001, 2003) studies logical relations among obligations and permissions, and more in particular violations and contrary-to-duty obligations, permissions and their relation to obligations, and the dynamics of obligations over time. Therefore, insights from deontic logic can be used to represent and reason with norms. Deontic logic also offers representations of norms as rules or conditionals. However, there are several aspects of norms which are not covered by constraints nor by deontic logic, such as the relation between the cognitive abilities of agents and the global properties of norms.

Conte et al. (1998) say that normative multiagent systems research focuses on two different sets of problems. On the one hand, they claim that legal theory and deontic logic supply a theory of norm-governed interaction of autonomous agents while at the same time lacking a model that integrates the different social and normative concepts of this theory. On the other hand, they claim that three other problems are of interest in multiagent systems research on norms: how agents can acquire norms, how agents can violate norms, and how an agent can be autonomous. For artificial agents, norms can be designed as in legal human systems, forced upon, for example when joining an institution, or they can emerge from the agents making them norm autonomous (Verhagen 1999). Agent decision making in normative systems and the relation between desires and obligations has been studied in agent architectures (Broersen et al. 2002), which thus explain how norms and obligations influence agent behavior.

An important question is where norms come from. Norms are not necessarily created by a single legislator, they can also emerge spontaneously, or be negotiated among the agents. In electronic commerce research, for example, cognitive foundations of social norms and contracts are studied (Boella and van der Torre 2006a). Protocols and social mechanisms are now being developed to support such creations of norms in multiagent systems. When norms are created, the question can be raised how they are enforced. For example, when a contract is violated, the violator may have to pay a penalty. But then there has to be a monitoring and sanctioning system, for example police agents in an electronic institution. Such protocols or roles in a multiagent system are part of the construction of social reality, and Searle (1995) has argued that such social realities are constructed by constitutive norms. This again raises the question how to represent such constitutive or counts-as norms, and how they are related to regulative norms like obligations and permissions (Boella and van der Torre 2006b).

Not only the relation between norms and agents must be studied, but also the relation between norms and other social and legal concepts. How do norms structure organizations? How about the relation between legal courts? Though in some normative multiagent systems there is only a single normative system, there can also be several of them, raising the question how normative systems interact. For example, in a virtual community of resource providers each provider may have its own normative system, which raises the question how one system can authorize access in another system, or how global policies can be defined to regulate these local policies (Boella and van der Torre 2006b).

3 Permissions and hierarchical normative systems

3.1 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. Strong permission only is a norm-character." (von Wright 1963 p. 86). Bobbio (1980) 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 norms 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 preexisting because a law itself may prescribe a limit to its own extension)", pp. 891–892.[1]

Moreover, Bobbio (1958) 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*."

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" (Bulygin 1986, p. 213).

3.2 Interaction games in hierarchical normative systems

The notion of permission alone is not enough to build a normative system, as Lewis (1979)'s "master and slave" game has shown: for only obligations 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".

Bulygin (1986) challenges the view that permission is only the lack of obligation: in a dynamic view of the normative system, illustrated by the games between "Rex" and "Subject", it is sensible to use permissions to derogate to former prohibitions. In this game, however, it is pointless to issue permissions which do not refer to a preexisting obligation.

Thus, Bulygin (1986) devices a new game, showing the role of permissions which do not refer to a preexisting prohibition: "suppose that Rex, tired of governing alone, decides one day to appoint a Minister and to endow him with legislative power. [...] an action commanded by Minister becomes as obligatory as if it would have been commanded by Rex. But Minister has no competence to alter the commands and permissions given by Rex." If Rex permits hunting on Saturday and then Minister prohibits it for the whole week, its prohibition on Saturday remains with no effect.

Summarizing, Bulygin explains why permissive norms are not superfluous even if there is no corresponding prohibition. What was lacking 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, as the "Rex, Minister and Subject" game shows. 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.

3.3 Permissions in hierarchical legal systems

Also Alchourron and Makinson (1981) 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.

[1] Authors' translation from Italian text.

Once we have a plurality of authorities, we need to cope with conflicts: the solution is to use *meta-norms*. The meta-norms of the normative 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” (Guastini 1998, p. 29), “lex posterior derogat priori” the function of abrogating preexisting imperative norms or to derogate to them, etc.

4 Input/output logic

Makinson and van der Torre (2000) see a normative code as a set G of conditional norms, which is a set of 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. Moreover, given any universe L such that $G \subseteq L^2$ and an input $A \subseteq L$, they suggest that the output of A under G may be understood simply as

$$G(A) = \{x \mid (a,x) \in G \text{ for some a} \in A\}$$

Input/output logic investigates what happens to this basic picture when we pass to the logical level, i.e., when L is a propositional language, closed under at least the usual truth-functional connectives, and G a set of ordered pairs (a, x) of formulae in L. Since this investigation is relevant not only for deontic logic, Makinson and van der Torre refer to G not as a normative code, but as a generating set. To avoid all confusion, the generators G are not treated as formulae, but simply as ordered pairs (a, x) of purely boolean (or eventually first-order) formulae. They read a pair (a, x) forwards, i.e., with a as body and x as head; and they call the corresponding truth-functional formula $a \rightarrow x$ its materialization, echoing the old name material implication for the connective involved.

Suppose that we are also given a set A of formulae. The problem studied in input/output logic is: how may we reasonably define the set of propositions x making up the output of A under G, or one might also say, of G given A, which we write $out(G,A)$? Makinson and van der Torre emphasize that 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’.

4.1 Obligations

The basic intuition is that input and output are both under the sway of the operation Cn of classical consequence. Makinson and van der Torre’s simplest response to their problem is to put

$$out(G, A) = Cn(G(Cn(A)))$$

where the function $G(.)$ is defined as on the pre-logical level above, and Cn alias $\vdash$ is classical consequence. In other words, given a set A of formulae as input, they first collect all of its consequences, then apply G to them, and finally consider all of the consequences of what is thus obtained. They also define various variants to deal with disjunctive inputs intelligently, and making outputs available for recycling as inputs.

Definition 1 (*Obligations*) (*Makinson and van der Torre* 2000) Let L be a propositional 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 x'. The following logical systems have been defined, where v ranges over boolean valuations or the function that puts $v(b) = 1$ for all formulae b, and $V = \{b \mid v(b) = 1\}$.

$out_1(G, A) = Cn(G(Cn(A)))$,

$out_2(G, A) = \cap\{Cn(G(V)) \mid v(A) = 1\}$,

$out_3(G, A) = \cap\{Cn(G(B)) \mid A \subseteq B = Cn(B) \supseteq G(B)\}$,

$out_4(G, A) = \cap\{Cn(G(V)) \mid v(A) = 1 \text{ and } G(V) \subseteq V\}$.

The following example illustrates and compares the four input/output logics. The most characteristic property is that inputs are not in general outputs; that is, we do not have $A \subseteq out_1(G, A)$.

Example 1 (Makinson and van der Torre 2000) Put generators $G = \{(a, x), (b, x), (x, y)\}$, where a, b, x and y are distinct elementary letters, and put $A = \{a\}$. Inputs are not in general outputs, since $G(Cn(\{a\})) = \{x\}$ so $a \notin out_1(G, \{a\}) = Cn(G(Cn(\{a\}))) = Cn(\{x\})$. Contraposition also fails, for although $x \in out_1(G, \{a\})$ we have $\neg a \notin out_1(G, \{\neg x\})$: since $a \notin Cn(\{\neg x\})$ we have $G(Cn(\{\neg x\})) = \emptyset$ so that $\neg a \notin out_1(G, \{\neg x\}) = Cn(G(Cn(\{\neg x\}))) = Cn(\emptyset)$.

We do not have $y \in out_1(G, A)$. However, in certain situations, it may be appropriate for outputs to be available for recycling as inputs. For example, the elements (a, x) of G may be conditional norms of a kind that say that any configuration in which a is true is one in which x is desirable. In some contexts, we may wish to entertain hypothetically the items already seen as desirable, in order to determine what is in turn so. We do have $y \in out_3(G, A)$ and $y \in out_4(G, A)$.

Finally, put $A = \{a \vee b\}$. Then $Cn(A) \cap b(G) = \emptyset$ where we write $b(G)$ for the set of all bodies of elements of G, i.e., in this example the set $\{a, b, x\}$. Hence also $G(Cn(A)) = \emptyset$ so that $out_1(G, A) = Cn(G(Cn(A))) = Cn(\emptyset)$. However, in many contexts we would want to put x in the output, as it can be obtained from each of the two disjuncts of the input. We do have $x \in out_2(G, A)$ and $x \in out_4(G, A)$.

Input/output logic is axiomatized as a kind of conditional logic, where one is used to ask the following question. Suppose we are given only the generating set G: how may we define the set of input/output pairs (a, x) arising from G, written as $out(G)$? Makinson and van der Torre suggest that this is the same question as asking what is $out(G, A)$, because they define $(\wedge A, x) \in out(G)$ iff $x \in out(G, A)$ for finite A. They also suggest that the two formulations give a rather different gestalt, and one is sometimes more convenient rather than the other. Whereas the latter tends to be clearer in semantic contexts, the former is easier to work with when considering derivations in a syntactic context. They move freely from one to the other, just as one moves between *Cn* and $\vdash$ for classical consequence.

Theorem 1 (Makinson and van der Torre 2000) *Let L be a base logic with* $\top$ *a tautology, and let G be a set of ordered pairs of L (called the generators). Input/output logic* $out_1(out_2/out_3/out_4)$ *is a closure operation on* $G \cup \{(\top, \top)\}$ *under replacement of logical equivalents in input or output and the rules SI, WO and AND (together with OR/CT/OR and CT).*

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

Example 2 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).

4.2 Permissions

Permissions are more ambiguous than obligations, and various notions have been defined. Makinson and van der Torre (2003) distinguish three notions of permission. First, *negperm* is the negation of an prohibition, 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. If P is the set of permissive norms, then we have $statperm(P,G) \subseteq out(P \cup G)$, see Makinson and van der Torre (2003) 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" (Makinson and van der Torre 2003).

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.

- $(a,x) \in negperm(G)$ *iff* $(a, \neg x) \notin out(G)$;
- $(a,x) \in statperm(P,G)$ *iff* $(a,x) \in out(G \cup Q)$ for some singleton or empty $Q \subseteq P$;

- $(a, x) \in \mathit{dynperm}(P, G)$ *iff* $(c, \neg z) \in \mathit{out}(G \cup \{(a, \neg x)\})$ for some pair $(c, z) \in \mathit{statperm}(P, G)$ with c consistent.

Example 3 It is obligatory to make homework, but if one does homework he is permitted to watch the television, $G = \{(\top, h)\}$ and $P = \{(h, w)\}$. Then $(\top, h) \in \mathit{negperm}(G)$, since what is obligatory is permitted and $(a, b) \in \mathit{negperm}(G)$ since given a there is no restriction about b. Moreover, $(h, w) \in \mathit{statperm}(P, G)$ since this is explicitly permitted and $(a, w) \in \mathit{dynperm}(P, G)$: $(a \wedge h, \neg w) \in \mathit{out}(G \cup \{(a, \neg w)\})$ for $(a \wedge h, w) \in \mathit{statperm}(P, G)$.

4.3 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 (Makinson and van der Torre 2001).

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 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" (Hansson 1969).

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 sets 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 consider only the input/output constraints.

Definition 3 (*Constraints*) Let G be a set of generators and *out* be an input/output logic. We define:

- $\mathit{maxfamily}(G, A)$ is the set of $\subseteq$-maximal subsets G' of G such that $\mathit{out}(G', A) \cup A$ is consistent.
- $\mathit{outfamily}(G, A)$ is the output under the elements of *maxfamily*, i.e., $\{\mathit{out}(G', A) \mid G' \in \mathit{maxfamily}(G, A)\}$.
- $(a, x) \in \mathit{out}_{\cup}(G)$ *iff* $x \in \cup\, \mathit{outfamily}(G, A)$ and $a = \wedge A_0$ for some finite $A_0 \subseteq A$
- $(a, x) \in \mathit{out}_{\cap}(G)$ *iff* $x \in \cap\, \mathit{outfamily}(G, A)$ and $a = \wedge A_0$ for some finite $A_0 \subseteq A$

Makinson and van der Torre (2001) consider the following example.

Example 4 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 $a \wedge \neg x$. On the one hand, $out(G, \{a \wedge \neg x\}) = Cn(\{\neg a, x, y\})$, 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.

Second, the three notions of permission do not cover permissions as exceptions, while, as we have seen in Sect. 3 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.

5 Priorities and hierarchy

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.

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 at least as preferred as A".

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

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

Example 5 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.

$\mathit{maxfamily}(G, P, V, \{s\}) = \{\{a, b\}, \{b, c\}\}$,
$\mathit{preffamily}(G, P, V, \leq, \{s\}) = \{\{b, c\}\}$,
$\mathit{outfamily}(G, P, V, \leq, \{s\}) = \{Cn(\{\top\})\}$
$\mathit{statpermfamily}(G, P, V, \leq, \{s\}) = \{Cn(\{k\})\}$

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, c is not applicable in situation s whereas 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 of the consequences of applying the generators whose pointer is in G which are included in $\{b, c\}$ and 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 (Alchourron and Makinson 1981), which we formalize as an ordering on the generator pointers, in which each cluster represents an authority.

5.1 Hierarchy

Alchourròn and Makinson (1981) define a hierarchy of regulations 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", (p. 126). Moreover, "[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", (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 Alchourròn and Makinson (1981) notice, "there is no *a priory* way of determining what the most suitable definitions will be".

So we adopt the following definition of lifting, inspired by Brass (1993): "$E_1 \leq E_2 \iff$ 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 $b \npreceq a$.

Example 6 The hierarchy $a \prec c \prec b$ respects the relation on set of rules used in Example 5 $\{a, c\} < \{a, b\} < \{b, c\}$. The norm 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 $a \preceq c$.

6 Static and dynamic norms

We now consider the distinction between static and dynamic norms. We say that a norm is weakly redundant, when the output of a given set of norms does not change when we remove the norm. We say that a norm is strongly redundant when the norm is weakly redundant for any extension of the set of norms.

6.1 Static norms

The static interpretation makes sense if the 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 $\forall a \in L$: $\mathit{outfamily}(G, P, V, \leq, \{a\}) = \mathit{outfamily}(G \setminus \{g\}, P \setminus \{g\}, V, \leq, \{a\})$.

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 *maxfamily*; adding a permission which does not affect the material validity of some obligation does not affect *outfamily*. In the same way as permissions,

[2] Where with respect to (Brass 1993) we reverse the meaning of the $\preceq$ relation: for Brass $B \leq C$ means that B is preferred over C.

[3] Alternative notions can be based on $\mathit{out}_{\cup/\cap}$. We do not detail the changes in $\leq$ and V, which have to shrink or grow in the obvious way with the removal or addition of norms. See Brewka and Eiter (2000) for a definition of agreement between two priority relations.

obligations can have a role of derogating some permission (as in Example 5) or even another obligation, as the following example shows:

Example 7 $G = \{a = (\top, \neg f), b = (p, f)\}, P = \emptyset$, $\{a\} < \{b\}$. No one should have firearms, policemen should have one.

$maxfamily(G, P, V, \{p\}) = \{\{a\}, \{b\}\}$,
$preffamily(G, P, V, \leq, \{p\}) = \{\{b\}\}$,
$outfamily(G, P, V, \leq, \{p\}) = \{Cn(\{f\})\}$,
$statpermfamily(G, P, V, \leq, \{p\}) = \{Cn(\{f\})\}$

6.2 Dynamic norms

We say that a norm is strongly redundant when it is weakly redundant for any extension of the set of norms. The new norms may have any priority, and the priority relation among the old norms remains unchanged.

Definition 7 (*Dynamic norms*) If $G \cup P$ is a set of norms extended with $G' \cup P'$, then we say that $\leq'$ extends $\leq$ if $\leq \subseteq \leq'$ and for all $g_1 \leq' g_2$ without $g_1 \leq g_2$, we have $g_1 \in G' \cup P'$ or $g_2 \in G' \cup P'$.

The norm $g \in G \cup P$ is strongly redundant if and only if $\forall a \in L, \forall G', P'$ and $\leq'$ extending $\leq$, we have $outfamily(G \cup G', P \cup P', V, \leq', \{a\}) = outfamily(G \setminus \{g\} \cup G', P \setminus \{g\} \cup P', V, \leq', \{a\})$.

Intuitively, an authority may introduce a weakly but not strongly redundant norm to block the possibility that lower level authorities introduce conflicting and materially valid norms. Roughly, norms are strongly redundant when they are logically implied, i.e., derived by the input/output logic. To model the scenarios of Bulygin and Lewis, we further detail the model of hierarchical normative systems by making the authorities and their competence explicit. In such a setting, we may say that the lower and higher levels of authorities play a game against each other.

In the Bulygin/Lewis games, the hierarchy on rules is due to the "lex superior" principle. Other principles are discussed in Prakken and Sartor (1996). These principles play the roles of *meta-norms* which, according to Mazzarese (1991), "establish which norms do constitute a given legal order", i.e., in our terminology, meta-norms establish which norms are materially valid.[4]

Definition 8 (*Competence and formal validity*) Let A be a set of authorities, and $\succeq_A$ an ordering on A, $aut : G \cup P \rightarrow A$ a function that associates an authority with each rule, and $C : A \rightarrow 2^L$ the competence of authority expressed by a set of propositional formulas of L. We say that:

- The hierarchy $\preceq$ reflects $\preceq_A$ if and only if $aut(g_1) \prec_A aut(g_2)$ implies $g_1 \prec g_2$.
- The normative system respects the competence of the authorities if and only if for each norm $g = (x, y) \in G \cup P$ we have that $y \in C(aut(g))$.

[4] In contrast with the notion of systemic validity of Mazzarese (2000), 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.

- $g \in G \cup P$ is strongly redundant with respect to $\langle A, aut, \preceq_A, C\rangle$ if and only if it is strongly redundant for all normative systems respecting the competence of the authorities.

The following example illustrates the static/dynamic distinction.

Example 8 (Continued from Example 5) 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, $a_1 \prec_A a_2$ 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.

The latter definition of strong redundancy supports Bulygin's argument in favor of permissions which do not presuppose explicitly existing obligations, because 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.

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:

$$\begin{aligned}
&\mathit{maxfamily}(G, P, V, \{q \wedge s\}) = \{\{a, b, e\}, \{a, b, d\}, \{b, c, d\}, \{b, c, e\}\},\\
&\mathit{preffamily}(G, P, V, \le, \{q \wedge s\}) = \{\{b, c, d\}\},\\
&\mathit{outfamily}(G, P, V, \le, \{q \wedge s\}) = \{Cn(\{r, p, \neg t\})\}.
\end{aligned}$$

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. 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 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 $\neg d \in \mathit{outfamily}(\{(c, \neg d)\}, \{(a, d)\}, V, \leq, \{c\})$.

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 $\top$):

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

7 Related work

Alchourròn and Makinson (1981) 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 $A \setminus D$ 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 et al. (1985).

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 (1997) 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."

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 (1987) 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 a argument $P \rightarrow Q$, an undercutter is a reason for $\neg(P \rightarrow Q)$. However, when we consider conditionals concerning norms, as input/output logic, we should not speak of their truth. So Pollock (1987)'s definition is not adequate for obligations and permissions.

More recently, Prakken (1997) 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 undercutted by an argument supporting R.

Verheij (2001) criticizes this approach since he argues that it limits the possibility to add further undercutting arguments once the conditional has been created. Hence Verheij (2001) 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. This allows to model the sequential character of law making, where new permissions can be added after the obligation they are exceptions to. Our model offers the same flexibility, with the restriction that we model only undercutters of formulas instead of defeating conditionals.

8 Summary

Normative multiagent systems study general and domain independent properties of norms. It builds on results obtained in deontic logic, the logic of obligations and permissions, for the representation of norms as rules, the application of such rules, contrary-to-duty reasoning and the relation to permissions. However, it goes beyond logical relations among obligations and permissions by explaining the relation among social norms and obligations, relating regulative norms to constitutive norms, explaining the evolution of normative systems, and much more.

The dynamic arrival of new issues and the need to introduce new norms to regulate them is a crucial problem for multiagent systems becoming open and more and more autonomous. Moreover, in such systems social laws are not only designed off-line, but they are also developed at run-time. In such systems, it is of crucial importance that the user still can control the development of social laws by restricting them in advance.

In this paper, we study institutions with hierarchies of authorities. Higher authorities in the hierarchy should delimit the legislative autonomy of lower ones. In particular, we study the interplay between obligations and permissions in the (Bulygin 1986)'s "Rex, Minister and Subject" games.

To study permissions and obligations we use 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 1986)'s 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). Without permissions as exceptions "there would be no possibility of normative change emerging from acts of authority" (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 (Boella and van der Torre, 2006a, b), extending Boella and Lesmo (2002)'s proposal in a qualitative decision theory, substantive and procedural norms (Boella and van der Torre, to appear), and negotiation of norms (Boella and van der Torre, 2007). Future work concerns the definition of institution with hierarchies of authorities using the sanction-based definition of norms.

References

Alchourron C, Gärdenfors P, Makinson D (1985) On the logic of theory change. J Symbol Logic 50(2):510–530

Alchourron CE, Makinson D (1981) Hierarchies of regulations and their logic. In: Hilpinen R (ed) New studies in deontic logic. D. Reidel, Dordrecht, pp 125–148

Bobbio N (1958) Teoria della norma giuridica. Torino, Giappichelli

Bobbio N (1980) Norma. In: *Enciclopedia Einaudi*, Vol. 9. Torino, Einaudi, pp 876–907

Boella G, Lesmo L (2002) A game theoretic approach to norms. Cogn Sci Quart 2(3–4):492–512

Boella G, van der Torre L (2006a) A Game Theoretic Approach to Contracts in Multiagent Systems. IEEE Trans Syst Man Cybernet Part C Appl Rev 36(1):68–79

Boella G, van der Torre L (2006b) Security policies for sharing knowledge in virtual communities. IEEE Trans Syst Man Cybernet Part A Syst Humans 36(3):439–450

Boella G, van der Torre L (2007) Norm negotiation in multiagent systems. Int J Coop Inf Syst 16(1):97–122

Boella G, van der Torre L (to appear) Substantive and procedural norms in normative multiagent systems. J Appl Logic

Boella G, van der Torre L, Verhagen H (2006) Introduction to normative multiagent systems. Comput Math Org Theory 12(2–3):71–79

Brass S. (1993) Deduction with supernormal defaults. In: Brewka G, Jantke KP, Schmitt PH (eds) Nonmonotonic and inductive logics, Vol. 659 of LNAI. Berlin, Springer, pp 153–174

Brewka G, Eiter T (2000) Prioritizing default logic. In: Intellectics and computational logic. Kluwer, pp 27–45

Broersen J, Dastani M, Hulstijn J, van der Torre L (2002) Goal generation in the BOID architecture. Cogn Sci Quart 2(3-4):428–447

Bulygin E (1986) Permissive norms and normative systems. In: Martino A, Natali FS (eds) Automated analysis of legal texts. Amsterdam, Publishing Company, pp 211–218

Bulygin E (1996) Logische Fragen der Gesetzgebungstechnik. In: Rodig J (ed) Studien zu einer Theorie der Gesetzgebung. Berlin, Springer, pp 612–627

Conte R, Castelfranchi C, Dignum F (1998) Autonomous norm-acceptance. In: Intelligent agents V (ATAL '98), Vol. 1555 of LNCS. Berlin, Springer, pp 99–112

Dignum F. (1999) Autonomous agents with norms. Artif Intel Law 7:69–79

Dignum V, Vázquez-Salceda J, Dignum F (2004) A model of almost everything: norms, structure and ontologies in agent organizations. In: Procs of 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04), New York (NJ), ACM, pp 1498–1499

Esteva M, Padget J, Sierra C (2001) Formalizing a language for institutions and norms. In: Intelligent agents VIII (ATAL'01), Vol. 2333 of LNCS. Berlin, Springer, pp 348–366

Esteva M, Rodriguez-Aguilar J, Sierra C, Vasconcelos W (2004) Verifying norm consistency in electronic institutions. In: Procs of Workshop on Agent Organizations at AAAI'04. San Jose (CA)

Guastini R (1998) Teoria e dogmatica delle fonti. Milano, Giuffrè

Hansson B (1969) An analysis of some deontic logics. Nous 3:373–398

Jones A, Carmo J (2001) Deontic logic and contrary-to-duties. In: Gabbay D, Guenthner F (eds) Handbook of philosophical logic, Vol. 3, Kluwer, Dordrecht (NL), pp 203–279

Lewis D (1979) A problem about permission. In: Saarinen E (ed) Essays in honour of Jaakko Hintikka. Dordrecht, D. Reidel, pp 163–175

Lopez y Lopez F, Luck M, d'Inverno M (2002) Contraining autonomy through norms. In: Procs of 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02), New York (NJ), ACM, pp 674–681

Makinson D, van der Torre L (2000) Input–output logics. J Philos Logic 29(4):383–408

Makinson D, van der Torre L (2001) Constraints for input–output logics. J Philos Logic 30(2):155–185

Makinson D, van der Torre L (2003) Permissions from an input–output perspective. J Philos Logic 32(4):391–416

Mazzarese T (1991) Deontic logic as logic of legal norms: Two main sources of problems. Ratio Juris 4:374–392

Mazzarese T. (2000) Permesso forte e permesso debole: note a margine. Analisi e diritto.

Pollock JL (1987) Defeasible reasoning. Cogn Sci 11:481–518

Prakken H (1997) Logical tools for modelling legal argument. Dordrecht, Kluwer Academic Publisher

Prakken H, Sartor G (1996) A dialectical model of assessing conflicting arguments in legal reasoning. Artif Intel Law 4(3–4):331–368

Royakkers L, Dignum F (1997) The logic of enactment. In: Procs of Fourth International Conference on Artificial Intelligence and Law (ICAIL'97), ACM, p. 257

Searle J (1995) The construction of social reality. New York, The Free Press

Shoham Y, Tennenholtz M (1995) On social laws for artificial agent societies: off-line design. Artif Intel 73(1–2):231–252

van der Torre L (2003) Contextual deontic logic: normative agents, violations and independence. Annl Math Artif Intel 37(1–2):33–63

van der Torre L, Tan Y (1999) Contrary-to-duty reasoning with preference-based dyadic obligations. Annl Math Artif Intel 27(1–4):49–78

Verhagen H (1999) On the learning of norms. In: Procs of Modelling Autonomous Agents in a Multi-Agent World (MAAMAW '99)

Verheij B (2001) Legal decision making as dialectical theory construction with argumentations schemes. In: Procs of Eighth International Conference on Artificial Intelligence and Law (ICAIL'01), ACM, pp 225–226

von Wright GH (1951) Deontic logic. Mind 60:1–15

von Wright GH (1963) Norm and action. A logical inquiry. Routledge and Kegan Paul, London