

Combining Constitutive and Regulative Norms in Input/Output Logic

Xin Sun and Leendert van der Torre

Computer Science and Communication, University of Luxembourg

Abstract. In this paper we study three semantics to combine constitutive and regulative norms. In the first semantics, called the simple-minded semantics, the output of the constitutive norms are intermediate facts used as input for the regulative norms. The second method is called throughput, and adds the input of the constitutive norms to the intermediate facts. The third method is called reusable throughput, because it reuses the output of the regulative norms in the input of the constitutive norms. In addition, we refine these three so-called abstract semantics such that the obligations are labeled with the intermediate facts used to derive them. These explanations in the labels can be used for norm change, interpretation or defeasible argumentation. We present complete axiomatisations for the abstract and refined versions of the three semantics.

Key words: deontic logic, input/output logic, constitutive norms

1 Introduction

Constitutive norms are normative reasoning discussed in the handbook of deontic logic and normative systems [6], besides permissive norms, prima facie norms, and normative positions. They are usually contrasted with norms regulating the behavior of human beings by indicating which behaviors are obligatory, permitted and forbidden. Constitutive norms do not regulate actions or states-of-affairs, but rather they define new possible actions or states of affairs. In this paper we have little to say about constitutive norms themselves, and we refer the reader to the overview chapter of Grossi and Jones in the handbook [9]. Instead, we are interested in the *combination* of constitutive and regulative norms. As there are various ways to combine these two kinds of norms, and we believe none of them is perfect, this raises a new question. Besides choosing a logic for constitutive norms and a logic for regulative norms, the new question is:

Research question. Which combination method to choose for combining constitutive and regulative norms?

Our approach in this paper to answer this question is to define three ways of combining these two kinds of norms, and to axiomatize these combinations. As always, the axiomatization presents the characteristic properties of the combination methods, which can be used to choose the method appropriate for a particular application. Moreover, we

make as little commitments as possible about the representation of the norms. For example, constitutive norms are represented by count-as conditionals “ X counts as Y in the context Z ” [24, 12, 10], but there does not seem to be a consensus on the representation of the context. We therefore follow Lindahl and Odelstad [15] and Boella and van der Torre [1, 2] and abstract away from the context. We thus represent constitutive norms as rules “ X counts as Y .” We use the general input/output logic approach [17, 18] to represent both constitutive and regulative norms, and in particular we use a ‘minimal’ input/output logic recently introduced by Parent and van der Torre [20, 21].

The research question breaks down into the following four subquestions. First, how to axiomatize the simple-minded combination method, as used by Lindahl and Odelstad [15], visualized below in Figure 1? Here C and R are the set of constitutive and regulative norms respectively. A is a set of formulas representing the facts. I is another set of formulas representing the intermediate concepts derived by the fact A and constitutive norms C . These intermediate facts are the input of regulative norms R . O is the output of intermediate facts I together with regulative norms R . If we write $I(C, A)$ for the intermediate facts derived from the facts A using constitutive norms C , and $\bigcirc(R, I)$ for the obligations derived from the intermediate facts I using the regulative norms R , then we can represent their simple-minded combination method as $\ominus^*(R, C, A) = \bigcirc(R, I(C, A))$.

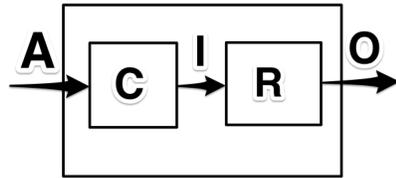


Fig. 1. Lindahl & Odelstad’s combination

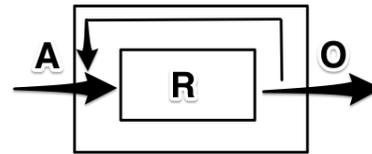


Fig. 2. Reusable input/output logic

Second, how to relate obligations explicitly to their intermediate facts, such that these can be used for explanation tasks in norm change, interpretation, and defeasible argumentation? For example, assume that a piece of paper counts as a contract, represented by the constitutive norm (*paper, contract*), and that the contract obliges us to pay money, represented by the regulative norm (*contract, pay*). From these two norms we want to derive the intermediate fact *contract* from the fact *paper*, and the obligation *paycontract*, which means we are obliged to pay because there is a contract. For example, in argumentation, we can present a rebutting argument there is no obligation to pay, or an undercutting argument that there is no contract. As another example, consider the well known story of *tû-tû* discussed by Ross [23]. Suppose (*eat, tû-tû*) represents “If a person has eaten the chief’s food, then she is *tû-tû*”, and (*tû-tû, purification*) represents “If a person is *tû-tû*, then she is obligatory to be subjected to a ceremony of purification.” Given the fact *eat*, from these two norms we can derive the institutional fact of *tû-tû* and the obligation *purification_{tû-tû}*, which means the person should be subjected to a ceremony of purification because she is *tû-tû*. Likewise, an obligation *congratulate_{checkmate}* says that you have to congratulate your opponent because you are checkmate in chess, and the obligation *takecarefamily_{married}* says that you have to

take care of your family because you are married. Ross argues that similar examples can be found everywhere in the law. Taxation law, for instance, provides an abundance of examples: it usually stipulates who is to count as a taxable subject, and it also provides numerous examples of mandatory norms predicated on the basis of these classifications in the form of tax liabilities.

Third, how to introduce the assumption that brute facts can be used to detach obligations from regulative norms too? Tosatto et al. [26] discuss the difference between $\ominus^*(R, C, A) = \bigcirc(R, I(C, A))$ and $\ominus(R, C, A) = \bigcirc(R, A \cup I(C, A))$. For example, given $R = \{(a, x), (p, y)\}$, $C = \{(b, p)\}$ and $A = \{a, b\}$, we have $\ominus^*(R, C, A) = \{y\}$ because $I(C, A) = \{p\}$ and $\bigcirc(R, \{p\}) = \{y\}$, and $\ominus(R, C, A) = \{x, y\}$ because $A \cup I(C, A) = \{a, b, p\}$ and $\bigcirc(R, \{a, b, p\}) = \{x, y\}$.

Fourth, how to ensure that the combined system has the same properties as the individual systems? For example, Parent and van der Torre [21] argue for a new form of deontic detachment, called aggregative deontic detachment. The corresponding rule for aggregative deontic detachment is called *aggregative cumulative transitivity* (ACT):

$$\text{(ACT)} \frac{x \in \bigcirc(R, a) \quad y \in \bigcirc(R, a \wedge x)}{x \wedge y \in \bigcirc(R, a)}$$

In this paper we use aggregative input/output logic to represent constitutive and regulative norms. The two combinations mentioned by Tosatto et al. [26] are therefore represented by $\bigcirc(R, \bigcirc(C, A))$ and $\bigcirc(R, A \cup \bigcirc(C, A))$, where the operator \bigcirc is defined by Parent and van der Torre [21]. We show in this paper that even though aggregative input/output logic satisfies ACT, both $\bigcirc(R, \bigcirc(C, A))$ and $\bigcirc(R, A \cup \bigcirc(C, A))$ do not. We therefore define a third combination of constitutive and regulative norms by forcing $\bigcirc(R, A \cup \bigcirc(C, A))$ to satisfy ACT. We call these three combinations simple-minded, throughput and reusable throughput combination respectively. For each of the three combinations, we further distinguish an *abstract combination*, where the output of the combination is a set of obligations, and a *detailed combination*, where the output of the combination is a set of obligations together with institutional facts.

Inspired by the input/output terminology, we use the following notation. We use \odot_1 , \odot_1^+ and \odot_3^+ for the semantics for the simple-minded, throughput and reusable throughput abstract combination respectively. The corresponding derivation system are represented by \triangleright_1 , \triangleright_1^+ and \triangleright_3^+ . For the detailed combinations, we use \odot_1 , \odot_1^+ and \odot_3^+ for semantics and \triangleright_1 , \triangleright_1^+ and \triangleright_3^+ for derivation system.

The layout of this paper is as follows. We survey aggregative input/output logic in Section 2. Then we introduce simple-minded, throughput and reusable combinations in Section 3 to 5 respectively. In Section 6 and 7 we discuss related work and future research.

2 Aggregative input/output logic

Parent and van der Torre [20, 21] introduce aggregative input/output logic, based on the following ideas. On the one hand, deontic detachment (DD) or cumulative transitivity (CT) is fully in line with the tradition in deontic logic. For instance, the Danielsson-Hansson-Lewis semantics [5, 11, 14] for conditional obligation validates such a law. On

the other hand, they also observe that potential counterexamples to DD may be found in the literature. Parent and van der Torre illustrate this with the following example, due to Broome [4, §7.4]:

You ought to exercise hard everyday
 If you exercise hard everyday, you ought to eat heartily
 ?* You ought to eat heartily

Intuitively, the obligation to eat heartily no longer holds if you take no exercise. Like the others, Parent and van der Torre claim that this counterexample suggests an alternative (they call it “aggregative”) form of detachment, which keeps track of what has been detached. They therefore reject the CT rule, and they accept the weaker ACT rule. As a consequence, and following an established tradition in the literature [7, 27, 25], weakening the output is no longer accepted either. In Parent and van der Torre [21], ACT is motivated as follows: “the counterexample usually given to CT in the literature no longer work when ACT is used in place of CT. This is because they all rely on the intuition that the obligation y ceases to hold when the obligation of (a, x) is violated”. Essentially the same argument is reproduced here by the example above. For another example consider the following:

Example 1. Consider the following situation:

- $(order, deliver)$ When you receive a purchase order from a customer, you have to deliver the good to the customer.
- $(order \wedge deliver, pay)$ If a customer sends a purchase order and the good are delivered to the customers, then the customer has the obligation to pay.
- $order \wedge \neg deliver$ The goods are ordered but not delivered.

Given the above situation, applying CT we derive $(order, pay)$. Then using factual detachment, which is a basic mechanism of input/output logic, we detach pay as our obligation. This conclusion is problematic, because intuitively we do not have to pay if the good is not delivered. On the contrary, applying ACT we can derive the argument $(order, deliver \wedge pay)$ but not $(order, pay)$. The problematic conclusion has disappeared.

Let $\mathbb{P} = \{p_0, p_1, \dots\}$ be a set of propositional letters and L be the propositional language built upon \mathbb{P} . We write $\phi \dashv\vdash \psi$ for logical equivalence in the logic L . Let R be a set of ordered pairs of formulas of L . A pair $(a, x) \in R$, call it a regulative norm, is read as “given a , it ought to be x ”. Let $A \subseteq L$, $R(A) = \{x \in L \mid (a, x) \in R, a \in A\}$ be set theoretically understood as the image of A under function R . The semantics of aggregative input/output logic is defined as following:

Definition 1 (Aggregative input/output logic [21]). For every $R \subseteq L \times L$, $A \subseteq L$, $x \in O(R, A)$ iff there is finite $R' \subseteq R$ with $R'(A) \neq \emptyset$ such that $\forall B = Cn(B)$, if $A \cup R'(B) \subseteq B$ then $x \dashv\vdash \bigwedge R'(B)$.

The above definition is partially visualized by Figure 2. In the definition there is a qualification over a logically closed set B , which represents the input of R . B is required to extend $A \cup R'(B)$ because the left arrow is labeled by A and there is influx from the right arrow (representing $R'(B)$) to the left arrow.

At a first glance it seems that such a reusable approach conflates the distinction between “ p is the case” and “ p is obligatory.” This is true in the sense that both facts and detached obligations are used as input to detach new obligations. Of course “ p is the case” and “ p is obligatory” are semantically different, but we believe such difference is not a sufficient reason to reject aggregative deontic detachment. For defense of aggregative deontic detachment the readers are suggested to consult the accompanying paper of Parent and van der Torre [21].

The following example illustrates aggregative deontic detachment.

Example 2. Let regulative norms be $R = \{(a, x), (a \wedge x, y)\}$ and input $A = \{a\}$. We have $O(R, A) = \{x, x \wedge y, \dots\}$. The table below illustrates how to calculate $O(R, A)$, where B^* is the smallest set of formulas such that we have $B^* = Cn(B^*)$ and moreover $A \cup R'(B^*) \subseteq B^*$. For $R = R'$ we can then derive the obligation for $x \wedge y$, but not an obligation for y , illustrating that ACT holds in aggregative input/output logic, but CT does not.

| A | R' | B^* | $R'(B^*)$ | $\wedge R'(B^*)$ |
|---------|--------------|-------------------|------------|-------------------------|
| $\{a\}$ | $\{(a, x)\}$ | $Cn(\{a, x\})$ | $\{x\}$ | $\{x, \dots\}$ |
| $\{a\}$ | R | $Cn(\{a, x, y\})$ | $\{x, y\}$ | $\{x \wedge y, \dots\}$ |

The proof system contains three rules: strengthening of the antecedent (SI), output equivalence (OEQ) and aggregative cumulative transitivity (ACT).

Definition 2 (Proof system of aggregative input/output logic [21]). Let $D(R)$ be the smallest set of arguments such that $R \subseteq D(R)$ and $D(R)$ is closed under the following rules:

- SI: from (a, x) and $b \vdash a$ to (b, x)
- OEQ: from (a, x) and $x \dashv\vdash y$ to (a, y)
- ACT: from (a, x) and $(a \wedge x, y)$ to $(a, x \wedge y)$.

The rule AND is derivable in aggregative input/output logic.

- AND: from (a, x) and (a, y) to $(a, x \wedge y)$

Parent and van der Torre define $x \in D(R, A)$ iff there exist $a_1, \dots, a_n \in A$ such that $(a_1 \wedge \dots \wedge a_n, x) \in D(R)$. The following completeness result is proved [21].

Theorem 1 (Completeness of aggregative input/output logic [21]). Given an arbitrary normative system R and a set A of formulas, $D(R, A) = O(N, A)$.

In this paper we use a variant of aggregative input/output logic: we delete the restriction of $R'(A) \neq \emptyset$ in Definition 1. Correspondingly, we require (\top, \top) to be included in the derivation system to ensure the soundness and completeness result.

Definition 3 (Variant of aggregative input/output logic). For every $R \subseteq L \times L$, $A \subseteq L$, $x \in \circ(R, A)$ iff there is finite $R' \subseteq R$ such that $\forall B = Cn(B)$, if $A \cup R'(B) \subseteq B$ then $x \dashv\vdash \wedge R'(B)$. $\triangleright(R)$ is the smallest set of arguments such that $\{(\top, \top)\} \cup R \subseteq \triangleright(R)$ and $\triangleright(R)$ is closed under the rules SI, OEQ and ACT.

Soundness and completeness of \circ and \triangleright follows the same lines as the soundness and completeness proofs of O and D .

3 Simple-minded combination

We use \circ not only for regulative norms, but also for constitutive norms. However, as constitutive norms cannot be violated, the issue raised by Parent and van der Torre does not occur for constitutive norms. In other words, we could have taken an input/output logic like reusable output or out_3 as well. The reason we choose to use \circ is uniformity, and in the detailed combination the derived obligations are more informative in the sense that the subscript contains *all* the institutional facts needed to derive the obligation.

The idea of simple-minded combination is illustrated by Figure 3. There is a set of constitutive norms $C \subseteq L \times L$ and a set of regulative norms $R \subseteq L \times L$. The input A is a set of formulas representing facts. $I = \circ(C, A)$ is the output produced by the semantics of aggregative input/output logic given C and A . I is understood as the intermediate facts and used as the input to regulative norms R to generate obligations $O = \circ(R, I)$.

We use aggregative input/output logic as our tool to analyze both constitutive and regulative norms. Since aggregative input/output logic is reusable in the sense its output can be reused as input, we represent simple-minded combination by Figure 3 with arrows representing reusability.

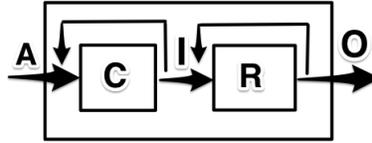


Fig. 3. Simple-minded combination

3.1 Simple-minded abstract combination

Simple-minded abstract combination can be built straightforwardly by a composition of two aggregative input/output logics.

Definition 4 (Semantics of simple-minded abstract combination). *Let C, R be two sets of constitutive and regulative norms respectively, the semantics of simple-minded abstract combination is:*

$$\odot_1(C, R, A) = \circ(R, \circ(C, A)).$$

Example 3. Let $A = \{eat\}$, $C = \{(eat, \hat{t}\hat{u}-\hat{t}\hat{u})\}$, $R = \{(\hat{t}\hat{u}-\hat{t}\hat{u}, purification), (eat, sorry)\}$. Here $(eat, sorry)$ means “if a person has eaten the chief’s food, then she should say sorry.” Then we have $\circ(C, A) = \{\hat{t}\hat{u}-\hat{t}\hat{u}, \dots\}$, $\odot_1(C, R, A) = \{purification, \dots\}$. Note we do not have $sorry \in \odot_1(C, R, A)$ because $eat \notin \circ(C, A)$.

The proof system of simple-minded abstract combination is base on the derivation system \triangleright for constitutive and regulative norms, with an additional composition.

Definition 5 (Derivation system of simple-minded abstract combination). *Let C, R be two sets of constitutive and regulative norms respectively, the proof system of simple-minded abstract combination is defined as follows:*

$$\triangleright_1(C, R) = \{(a, x) \mid \text{there is } p \in L \text{ such that } (a, p) \in \triangleright(C) \text{ and } (p, x) \in \triangleright(R)\}$$

We call the rule to derive $(a, x) \in \triangleright_1(C, R)$ from $(a, p) \in \triangleright(C)$ and $(p, x) \in \triangleright(R)$ abstract constitutive regulative transitivity (ACRT).

Example 4. From $C = \{(a, x), (a \wedge x, y)\}$, $R = \{(y, z)\}$ we can derive $(a, z) \in \triangleright_1(C, R)$ as following:

$$\frac{\frac{(a, x) \in C \quad (a \wedge x, y) \in C}{(a, x \wedge y) \in \triangleright(C)} \text{ (ACT)} \quad \frac{(y, z) \in R}{(x \wedge y, z) \in \triangleright(R)} \text{ (SI)}}{(a, z) \in \triangleright_1(C, R)} \text{ (DCRT)}$$

Like in the proof theory of aggregative input/output logic, we let $x \in \triangleright(C, R, A)$ iff there exist $a_1, \dots, a_n \in A$ such that $(a_1 \wedge \dots \wedge a_n, x) \in \triangleright(C, R)$. The semantics and proof theory of simple-minded abstract combination are connected by the following completeness result:

Theorem 2 (Completeness of simple-minded abstract combination). *Let C, R be two sets of constitutive and regulative norms respectively, and a a formula, we have*

$$x \in \odot_1(C, R, A) \text{ iff } x \in \triangleright_1(C, R, A).$$

The reader may be surprised about our choice of calling \triangleright_1 a proof system, whereas it has a semantical flavor, in the sense that it is defined by the composition of binary relations—which is a set-theoretic and not a syntactic operation. Consequently, the coincidence is not a classical completeness result in the sense of connecting a calculus with a set-theoretic construction. However, \triangleright_1 still is a kind of very simple proof system, building on derivations in the underlying logic \triangleright , and it is in spirit similar to the other proof systems used in the input/output logic framework and in this paper. We therefore prefer to call Theorem 2 an completeness rather than a representation result.

The following proposition shows some basic properties of simple-minded abstract combination.

Proposition 1. $\triangleright_1(C, R)$ validates SI, OEQ and AND, but not ACT.

3.2 Simple-minded detailed combination

In the semantics of aggregative input/output logic (Definition 2), we pick a set R' of the norms and qualify over a set of formulas B , which is closed under logical consequence. In the semantics of simple-minded detailed combination, we pick two sets C' and R' , and we qualify over two sets of formulas B_1, B_2 , which are both closed under logical consequence. The set B_1 is the input for C' . As visualized in Figure 3, we require it to extend $A \cup C'(B_1)$ because there is an arrow labelled A inject to C' and there is another arrow, the arrow from I to A , also inject to C' . Here note that I is $C'(B_1)$. Similarly, the set B_2 is the input for R' . We require it to contain $C'(B_1) \cup R'(B_2)$ because there is an arrow labeled I inject to R and there is another arrow, the arrow from O to I , also inject to R . Here note that O is $R'(B_2)$.

For the detailed combinations, we want to produce not only an obligation, but an obligation together with institutional facts. Formally, the output for the detailed combination is of the form x_p , where $x, p \in L$. As far as we know, such mixed output has not been defined in the input/output logic literature yet. Technically, the semantics of simple-minded detailed combination is defined as follows, in a flavor similar to aggregative input/output logic:

Definition 6 (Semantics of simple-minded detailed combination). *Let C, R be two sets of constitutive and regulative norms respectively, we define $x_p \in \odot_1(C, R, A)$ iff there is finite $C' \subseteq C, R' \subseteq R$ such that for all $B_1 = Cn(B_1), B_2 = Cn(B_2)$, if $A \cup C'(B_1) \subseteq B_1$ then $p \dashv\vdash \bigwedge C'(B_1)$, if $C'(B_1) \cup R'(B_2) \subseteq B_2$, then $x \dashv\vdash \bigwedge R'(B_2)$.*

Definition 7 (Proof theory of simple-minded detailed combination). *Let C, R be two sets of constitutive and regulative norms respectively, the proof theory of simple-minded detailed combination is:*

$$\blacktriangleright_1(C, R) = \{(a, x_p) \mid \text{there is } p \in L \text{ such that } (a, p) \in \triangleright(C) \text{ and } (p, x) \in \triangleright(R)\}$$

We call the rule deriving $(a, x_p) \in \blacktriangleright_1(C, R)$ from $(a, p) \in \triangleright(C)$ and $(p, x) \in \triangleright(R)$ detailed constitutive regulative transitivity (DCRT).

Example 5 illustrates that the subscript contains *all* the institutional facts needed to derive the obligation.

Example 5 (continued). From $C = \{(a, x), (a \wedge x, y)\}$, $R = \{(y, z)\}$ we can derive $(a, z_{x \wedge y}) \in \blacktriangleright_1(C, R)$ as follows:

$$\frac{\frac{(a, x) \in C \quad (a \wedge x, y) \in C}{(a, x \wedge y) \in \triangleright(C)} \quad (ACT) \quad \frac{(y, z) \in R}{(x \wedge y, z) \in \triangleright(R)} \quad (SI)}{(a, z_{x \wedge y}) \in \blacktriangleright_1(C, R)} \quad (DCRT)$$

In other words, to derive the obligation for z we need the institutional facts x and y .

With the proof theory and semantics as defined, we have the following completeness result.

Theorem 3 (Completeness of simple-minded detailed combination). *Let C, R be two sets of constitutive and regulative norms respectively,*

$$x_p \in \blacktriangleright_1(C, R, A) \text{ iff } x_p \in \odot_1(C, R, A).$$

The above proof theory relies heavily on the proof theory of aggregative input/output logic. Moreover, it works separately on constitutive and regulative norms and combines them together at the last step by the DCRT rule. We alternatively define an equivalent proof theory more directly on expressions of the form (a, x_p) .

Definition 8 (Alternative proof theory of simple-minded detailed combination). *Given C, R , let $\blacktriangleright'_1(C, R)$ be the smallest set of arguments such that $(\top, \top) \in \blacktriangleright'_1(C, R)$, $\{(a, \top_p) \mid (a, p) \in C\} \subseteq \blacktriangleright'_1(C, R)$ and $\blacktriangleright'_1(C, R)$ is closed under the following rules:*

- *SI: strengthening of the input: from (a, x_p) to (b, x_p) whenever $b \vdash a$*

- *IOEQ*: intermediate and output equivalence: from (a, x_p) to (a, y_q) if $p \dashv\vdash q$ and $x \dashv\vdash y$
- *ACTI*: aggregative cumulative transitivity for the intermediate: from (a, x_p) and $(a \wedge p, x_q)$ to $(a, x_{p \wedge q})$
- *ACTO*: aggregative cumulative transitivity for output: from (a, x_p) and $(a \wedge x, y_p)$ to $(a, x \wedge y_p)$,

and the following indexed constitutive/regulative transitivity (ICRT) rule:

- if $(a, \top_p) \in \triangleright'_1(C, R)$ and $(p, x) \in \triangleright(R)$ then $(a, x_p) \in \triangleright'_1(C, R)$.

Example 6. (continued) Given $C = \{(a, x), (a \wedge x, y)\}$, $R = \{(y, z)\}$, we first derive (a, \top_x) and $(a \wedge x, \top_y)$, then we derive $(a, z_{x \wedge y}) \in \triangleright'_1(C, R)$ as follows:

$$\frac{\frac{(a, \top_x) \in \triangleright'_1(C, R) \quad (a \wedge x, \top_y) \in \triangleright'_1(C, R)}{(a, \top_{x \wedge y}) \in \triangleright'_1(C, R)} \text{ (ACTI)} \quad \frac{(y, z) \in R}{(x \wedge y, z) \in \triangleright(R)} \text{ (SI)}}{(a, z_{x \wedge y}) \in \triangleright'_1(C, R)} \text{ (ICRT)}$$

The proof theory $\triangleright'_1(C, R)$ may look unusual at first glance, but it resembles the proof theory of aggregative input/output logic. They both contain rules like strengthening of the input, output equivalence and aggregative cumulative transitivity. The two derivation systems $\triangleright_1(C, R)$ and $\triangleright'_1(C, R)$ are equivalent.

Proposition 2. Let C, R be two sets of constitutive and regulative norms respectively, $\triangleright_1(C, R) = \triangleright'_1(C, R)$.

4 Throughput combination

In this section we strengthen simple-minded combination to throughput combination such that the input A can directly be used by regulative norms R , see Figure 4.

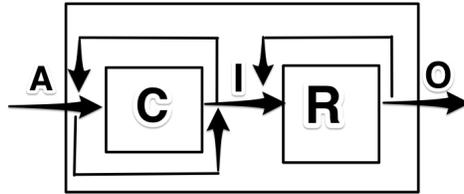


Fig. 4. Throughput combination

4.1 Throughput abstract combination

Throughput abstract combination is visualized by Figure 4, where both A and the output $\circ(C, A)$ are part of the input of $\circ(R, I)$.

Definition 9 (Semantics of throughput abstract combination). *Let C, R be two sets of constitutive and regulative norms respectively, $\odot_1^+(C, R, A) = \circ(R, A \cup \circ(C, A))$.*

The following is the proof system, in which the abstract constitutive regulative transitivity rule of simple minded combination is replaced by abstract constitutive regulative cumulative transitivity.

Definition 10 (Proof system of throughput abstract combination). *Let C, R be two sets of constitutive and regulative norms respectively, the proof system of throughput abstract combination is:*

$\triangleright_1^+(C, R) = \{(a, x) \mid \text{there is } p \in L \text{ such that } (a, p) \in \triangleright(C) \text{ and } (a \wedge p, x) \in \triangleright(R)\}$
We call the rule to derive $(a, x) \in \triangleright_1(C, R)$ from $(a, p) \in \triangleright(C)$, $(a \wedge p, x) \in \triangleright(R)$ abstract constitutive regulative cumulative transitivity (ACRCT).

Example 7. Given $C = \{(a, x), (a \wedge x, y)\}$ and $R = \{(a \wedge x \wedge y, z)\}$, we can derive $(a, z) \in \triangleright_1^+(C, R)$ as following:

$$\frac{\frac{(a, x) \in C \quad (a \wedge x, y) \in C}{(a, x \wedge y) \in \triangleright(C)} \quad (ACT) \quad (a \wedge x \wedge y, z) \in R}{(a, z) \in \triangleright_1^+(C, R)} \quad (ACRT)$$

Here note that $(a, z) \notin \triangleright_1(C, R)$ because $(x \wedge y, z) \notin \triangleright(R)$. It can be further proved that $\triangleright_1(C, R) \subseteq \triangleright_1^+(C, R)$

The semantics and proof theory of throughput abstract combination are connected by the following completeness result.

Theorem 4 (Completeness of throughput abstract combination). *Let C, R be two sets of constitutive and regulative norms respectively, we have $x \in \odot_1^+(C, R, a)$ iff $(a, x) \in \triangleright_1^+(C, R)$.*

4.2 Throughput detailed combination

In parallel to the simple-minded detailed combination, we introduce the semantics and proof theory of throughput detailed combination. The semantics of simple-minded detailed combination is similar to the semantics of aggregative input/output logic.

Definition 11 (Semantics of throughput detailed combination). *Let C, R be two sets of constitutive and regulative norms respectively, $A \subseteq L$, we define $x_p \in \odot_1^+(C, R, A)$ iff there is finite $C' \subseteq C$, $R' \subseteq R$ such that for all $B_1 = Cn(B_1)$, $B_2 = Cn(B_2)$, if we have $A \cup C'(B_1) \subseteq B_1$, then $p \dashv\vdash \wedge C'(B_1)$, if $A \cup C'(B_1) \cup R'(B_2) \subseteq B_2$, then we also have $x \dashv\vdash \wedge R'(B_2)$.*

Like the semantics of simple-minded detailed combination, here we pick two sets C' and R' , and we qualify over two sets of formulas B_1, B_2 , which are both closed under logical consequence. The only difference is that for B_2 , here we require it to extend A , while in simple-minded detailed combination we do not have such a requirement. The reason of this difference can be visualized by comparing Figure 3 and 4. In Figure 4 there is an arrow from A to I , while in Figure 3 there is not.

Definition 12 (Proof system of throughput detailed combination). *Let C, R be two sets of constitutive and regulative norms respectively, the proof system of throughput detailed combination is:*

$\triangleright_1^+(C, R) = \{(a, x_p) \mid \text{there is } p \in L \text{ such that } (a, p) \in \triangleright(C) \text{ and } (a \wedge p, x) \in \triangleright(R)\}$.
We call the rule to derive $(a, x_p) \in \triangleright_1^+(C, R)$ from $(a, p) \in \triangleright(C)$, $(a \wedge p, x) \in \triangleright(R)$ detailed constitutive regulative cumulative transitivity (DCRCT).

The proof system defined by Definition 12 and semantics are sound and complete.

Theorem 5 (Completeness of throughput detailed combination). *For all set of constitutive norms C , and regulative norms R , $(a, x_p) \in \triangleright_1^+(C, R)$ iff $x_p \in \odot_1^+(C, R, a)$.*

Like the proof system $\triangleright_1, \triangleright_1^+$ heavily relies on the proof system of aggregative input/output logic. A more independent proof system is defined as follows:

Definition 13 (Alternative proof system of throughput detailed combination). *Let C, R be two sets of constitutive and regulative norms respectively, Let $\triangleright_1^{+'}(C, R)$ be the smallest set such that $(\top, \top_\top) \in \triangleright_1^{+'}(C, R)$, $\{(a, \top_p) \mid (a, p) \in C\} \subseteq \triangleright_1^{+'}(C, R)$, $\{(a, x_\top) \mid (a, x) \in R\} \subseteq \triangleright_1^{+'}(C, R)$ and $\triangleright_1^{+'}(C, R)$ is closed under the rules SI, IOEQ, ACTI, ACTO and the following rule:*

– if $(a, \top_p) \in \triangleright_1^{+'}(C, R)$ and $(a \wedge p, x) \in \triangleright(R)$ then $(a, x_p) \in \triangleright_1^{+'}(C, R)$

One difference between $\triangleright_1^{+'}(C, R)$ and $\triangleright_1^+(C, R)$ is: for $\triangleright_1^{+'}(C, R)$ we require it to extend $\{(a, x_\top) \mid (a, x) \in R\}$. This feature reveals that regulative arguments can be derived directly in throughput combination. The equivalence of \triangleright_1^+ and $\triangleright_1^{+'}$ is stated in the following proposition.

Proposition 3. *For all $C, R \subseteq L \times L$, $(a, x_p) \in \triangleright_1^+(C, R)$ iff $(a, x_p) \in \triangleright_1^{+'}(C, R)$.*

5 Throughput reusable combination

Now we turn to throughput reusable combinations. As illustrated by the arrow from O to A in Figure 5, throughput reusable combination is the extension of throughput combination which allowing the output of regulative norms to be reused as input for constitutive norms. In this case the input of C have three resource: the arrow A , the arrow from I to A , and the arrow from O to A . The input of R have exactly the same resource. Therefore we can change Figure 5 to Figure 6 such that C and R have the same input.

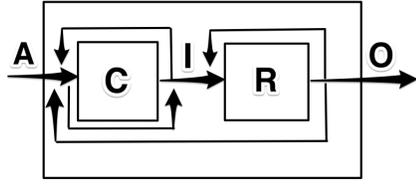


Fig. 5. Reusable combination

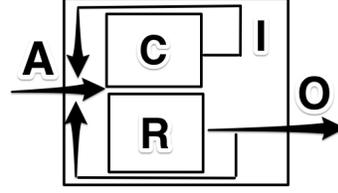


Fig. 6. Serial visualization

5.1 Throughput reusable abstract combination

The fact that throughput reusable combination is the extension of throughput combination which allows the reusability of output from R suggests that the former can be defined as the extension of latter validating the ACT rule. While the proof theory of throughput reusable abstract combination is a straightforward extension of its non-reusable companion, its semantics looks closer to the semantics of aggregative input/output logic.

Definition 14 (Semantics of throughput reusable abstract combination). Let C, R be two sets of constitutive and regulative norms respectively, and $A \subseteq L$, we define $x \in \odot_3^+(C, R, A)$ iff there exist finite $C' \subseteq C, R' \subseteq R$ such that for all $B = Cn(B)$: if $A \cup C'(B) \cup R'(B) \subseteq B$ then $x \dashv\vdash \bigwedge R'(B)$.

Definition 15 (Proof system of throughput reusable abstract combination). Let C, R be two sets of constitutive and regulative norms respectively, the proof system of throughput reusable abstract combination is defined as follows:

$\triangleright_3^+(C, R)$ is the smallest set such that $\triangleright_1^+(C, R) \subseteq \triangleright_3^+(C, R)$ and $\triangleright_3^+(C, R)$ is closed under the ACT rule.

The above semantics reflects the ideas illustrated by Figure 6. We qualify over a set B , which is the input for C and R . Such B is an extension of A , $I = C(B)$ and $O = R(B)$. We therefore require $A \cup C'(B) \cup R'(B) \subseteq B$. The semantics and proof theory of throughput reusable abstract combination are connected by the following completeness result:

Theorem 6 (Completeness of throughput reusable abstract combination). Let C, R be two sets of constitutive and regulative norms respectively, we have $x \in \odot_3^+(C, R, a)$ iff $(a, x) \in \triangleright_3^+(C, R)$.

5.2 Throughput reusable detailed combination

The semantics of throughput reusable detailed combination is an extension of its abstract companion.

Definition 16 (Semantics of throughput reusable detailed combination). Let C, R be two sets of constitutive and regulative norms respectively, and $A \subseteq L$, we define $x_p \in \odot_3^+(C, R, A)$ iff there exist finite $C' \subseteq C, R' \subseteq R$ such that for all $B = Cn(B)$, if $A \cup C'(B) \cup R'(B) \subseteq B$ then $p \dashv\vdash \bigwedge C'(B)$ and $x \dashv\vdash \bigwedge R'(B)$.

The following variant from an example of Makinson [16] illustrates that the combinations studied in this paper can also be used in other combination problems, not only for combining constitutive and regulative norms.

Example 8. Let $C = \{(A4, \text{size}), (t25 \times 15, \text{area})\}$, $R = \{(\text{size}, t25 \times 15), (\text{area}, \text{ref}10)\}$. Here “A4” means “the paper is an A4 paper”, “size” means “the paper is of standard size”, “t25×15” means “the text area is 25 by 15 cm”, “area” means “the paper is of standard text area”, “ref10” means “the font size for the reference is 10 points.” In this setting we have $\odot_3^+(C, R, \{A4\}) = \{\top, t25 \times 15, t25 \times 15 \wedge \text{ref}10, \dots\}$. For detailed combination we have $\odot_3^+(C, R, \{A4\}) = \{\top, t25 \times 15_{\text{size}}, t25 \times 15 \wedge \text{ref}10_{\text{size} \wedge \text{area}}, \dots\}$. The calculation can be illustrated by the following table, in which the column of A represents the input, C' and R' represent the subset of C and R , B^* represents the smallest set such that that $B^* = Cn(B^*)$ and $A \cup C'(B^*) \cup R'(B^*) \subseteq B^*$.

| A | C' | R' | B^* | $C'(B^*)$ | $R'(B^*)$ |
|-----|-------------------------|------------------------------------|---|--------------------------------|-----------------------------------|
| A4 | \emptyset | \emptyset | $Cn(\{A4\})$ | \emptyset | \emptyset |
| A4 | $\{(A4, \text{size})\}$ | $\{(\text{size}, t25 \times 15)\}$ | $Cn(\{A4, \text{size}, t25 \times 15\})$ | $\{\text{size}\}$ | $\{t25 \times 15\}$ |
| A4 | C | R | $Cn(\{A4, \text{size}, t25 \times 15, \text{area}, \text{ref}10\})$ | $\{\text{size}, \text{area}\}$ | $\{t25 \times 15, \text{ref}10\}$ |

In the second row of the table, $C'(B^*) = R'(B^*) = \emptyset$. This explains the reason of $\top \in \odot_3^+(C, R, \{A4\})$ and $\top \in \odot_3^+(C, R, \{A4\})$. The third row explains the reason for $t25 \times 15 \in \odot_3^+(C, R, \{A4\})$, $t25 \times 15_{\text{size}} \in \odot_3^+(C, R, \{A4\})$, and the fourth row for $t25 \times 15 \wedge \text{ref}10$.

Definition 17 introduces the proof system of throughput reusable detailed combination \triangleright_3^+ . It is an extension of \triangleright_1^+ , but its formation is simpler in the sense we add one rule called ACTIO but delete both ACTI and ACTO. Both ACTI and ACTO are derivable in \triangleright_3^+ .

Definition 17 (Proof system of reusable detailed combination). Let $\triangleright_3^+(C, R)$ be the smallest set such that $(\top, \top) \in \triangleright_3^+(C, R)$, $\{(a, x) \mid (a, x) \in R\} \subseteq \triangleright_3^+(C, R)$, $\{(a, \top_p) \mid (a, p) \in C\} \subseteq \triangleright_3^+(C, R)$ and $\triangleright_3^+(C, R)$ is closed under the rules SI, IOEQ, the following rule ACTIO

- ACTIO: aggregative cumulative transitivity for the intermediate and output: from (a, x_p) and $(a \wedge p \wedge x, y_q)$ to $(a, x \wedge y_{p \wedge q})$,

and the following rule

- if $(a, \top_p) \in \triangleright_3^+(C, R)$ and $(a \wedge p, x) \in \triangleright(R)$ then $(a, x_p) \in \triangleright_3^+(C, R)$.

Example 9 (continued). From $C = \{(A4, \text{size}), (t25 \times 15, \text{area})\}$, $R = \{(\text{size}, t25 \times 15), (\text{area}, \text{ref}10)\}$ we can derive expressions $(A4, \top_{\text{size}})$, $(t25 \times 15, \top_{\text{area}})$, $(\text{size}, t25 \times 15_{\top})$ and $(\text{area}, \text{ref}10_{\top})$. The following is the derivation:

$$\frac{\frac{(A4, \top_{\text{size}}) \quad \frac{(\text{size}, t25 \times 15_{\top})}{(A4 \wedge \top \wedge \text{size}, t25 \times 15_{\top})} \text{(SI)}}{(A4, t25 \times 15_{\top} \wedge \text{size}_{\top})} \text{(ACTIO)}}{(A4, t25 \times 15_{\text{size}})} \text{(IOEQ)}$$

Theorem 7 (Completeness of reusable detailed combination). *Given an arbitrary constitutive normative system C , regulative normative system R and a set A of formulas, $(a, x_p) \in \triangleright_3^+(C, R)$ iff $x_p \in \odot_3^+(C, R, a)$.*

6 Related work

Grossi and Jones [9] track the distinction between constitutive norms and regulative norms to at least Rawls [22]. Searle [24] uses ‘ X counts as Y in context C ’ as a canonical presentation of constitutive norms. Jones and Sergot [12] formalize the context as an institution. Grossi et al. [10] formalize the context as a set norms.

Grossi and Jones [9] present a classification of approaches to formalize constitutive norms, and we refer to their chapter for further background on the logic of constitutive norms. On the one hand the choice of logic of constitutive norms is orthogonal to the choice of combination method. On the other hand, we adopt aggregative input/output logic as our logic for constitutive norms, which is different from all the approaches summarized by Grossi and Jones [9].

Modalities can be combined using possible world semantics. Boutilier [3] introduces a model $M = (W, \geq_1, \geq_2, V)$ with \geq_1, \geq_2 total pre-orders over W , reflecting normality and preference respectively. Each pre-order defines a classical Danielsson-Hansson-Lewis dyadic operator \circ^i . Roughly, Boutilier defines a modality $\circ^{12}(B|A)$ as the best of the most normal A worlds satisfy B :

- $M, w \models \circ^i(B | A)$ iff $opt_{\geq_i}(\|A\|) \subseteq \|B\|$, for $i = 1, 2$
- $M, w \models \circ^{12}(B | A)$ iff $opt_{\geq_2}(opt_{\geq_1}(\|A\|)) \subseteq \|B\|$

Here $opt_{\geq_i}(S) = \{w \in S | \forall u \in S, w \geq_i u\}$, for every $S \subseteq W$. According to the semantics, the combined modality does not satisfy the combination properties discussed in this paper:

Observation 1 $\{\circ^1(q | p), \circ^2(r | q)\} \neq \circ^{12}(r | p)$

Lang and van der Torre [13] define $\circ^{12}(B | A)$ in the same models by: the most normal $A \wedge B$ is preferred to the most normal $A \wedge \neg B$, and they compare their definition with Boutilier’s. Observation 1 also holds for Lang and van der Torre’s combination. A further comparison between these modal logic approaches and our approach is left for further research.

7 Future research

We use aggregative input/output logic as our basis. Apart from the problem of pragmatic oddity and the irrelevant obligation problem mentioned in Parent and van der Torre [21], how to deal with the case when the output of constitutive norms is inconsistent is worthy of future research. The throughput reusable combination is formed by adding ACT to throughput combination. We can form a weaker version of reusable combination by adding ACT to simple-minded combination. The task is to define a semantics and prove the completeness result.

Makinson and van der Torre [17] developed input/output logic not only for obligations:

“In a range of contexts, one comes across processes resembling inference, but where input propositions are not in general included among outputs, and the operation is not in any way reversible. Examples arise in contexts of conditional obligations, goals, ideals, preferences, actions, and beliefs. Our purpose is to develop a theory of such input/output operations.”

Therefore, in future research we want to investigate whether our framework can be used also for combining other modalities. Consider the well known problem of combining beliefs and desires: you may desire to go to the dentist, you may believe that going to the dentist means that you will have pain, but you do not desire to have pain. We can model this in our framework, if C stands for beliefs (or knowledge) and R are desires (or obligations), then we have $C = (\text{dentist}, \text{pain})$ and $R = (\top, \text{dentist})$. In all the three abstract combinations we have $\text{dentist} \in \bigcirc_{*s}(C, R, \emptyset)$ but not $\text{pain} \in \bigcirc_{*s}(C, R, \emptyset)$. It has been argued also that side effects are important, and that we need to avoid only unwanted side effects [8].

Input/output logic can be translated into modal logic. Makinson and van der Torre have done such a translation in their original paper [17]. In our case, we can translate $(a, p) \in \triangleright(C)$ to $I_c(a) \rightarrow O_c(p)$, $(p, x) \in \triangleright(R)$ to $I_r(p) \rightarrow O_r(x)$. Here I_c , I_r , O_c and O_r are all modal operators. Moreover, we can translate $(a, x) \in \triangleright(C, R)$ to $I_c(a) \rightarrow O_r(x)$ and $(a, x_p) \in \blacktriangleright(C, R)$ to $I_c(a) \rightarrow O_c(p) \wedge O_r(x)$. How to give sound and complete representations of the logics and combination methods discussed in this paper using modal logic are problems to be solved in the future. We generalized input/output logic by considering two sets of norms. It can be further extended to LIONS, as foreseen by Makinson and van der Torre [19]. Moreover, to refer to the special topic of DEON14, it may be a first step towards a Kratzer style semantics of natural language, because Kratzer’s semantics can combine various kinds of ordering bases too.

8 Summary

To reason with constitutive and regulative norms, one has to choose a logic for the constitutive norms, a logic for the regulative norms, and a semantics to combine these two logics. In this paper we consider the question which semantics to choose for combining constitutive and regulative norms, a topic which has not raised much attention thus far, without committing ourselves to particular logics for constitutive or regulative norms. To make our analysis general, we use the ‘minimal’ logic introduced by Parent and van der Torre. Nevertheless, it contains two assumptions. First, strengthening of the input seems to reflect that rules do not have exceptions, whereas both constitutive and regulative norms encountered in practice often do have such exceptions. We do not consider this a limitation, because if we add priorities or a normality relation to reflect prima facie norms, exactly the same analysis can be given. Second, Parent and van der Torre’s logic satisfies aggregative deontic detachment. This is only a weak notion of deontic detachment, and some kind of deontic detachment is needed in the logics to be able to define the reusability semantics. We distinguished three semantics to combine constitutive and regulative norms:

The simple-minded combination is the least committed, and thus the safest one to use. It clearly distinguishes the input, intermediate facts and output obligations,

there is no possible source for confusion. It may be used, for example, when the input may not be true at the intermediate stage. For example, in legal interpretation the input may contain a bicycle which is also a vehicle, but the bicycle may not count as a vehicle in the legal sense. Using the simple-minded semantics, the intermediate facts may not contain a fact that the bicycle is a vehicle, whereas the input does. The proof system shows that the extension of the semantic is minimal, in the sense that it contains only a transitivity proof rule for the combination (Definition 4 and Theorem 2).

The throughput combination includes the input among the intermediate facts. The proof system shows that the difference with the simple-minded semantics is small, we just have to replace the transitivity axiom by a cumulative transitivity rule (Theorem 4). However, it means that for example in the bicycle case, we need to introduce another concept in the intermediate facts to represent that the bicycle does not count as a vehicle in the legal sense. In many common examples, it seems that the throughput semantics is preferred to the simple-minded semantics.

The throughput reusable combination considers as the intermediate state the facts closed under both the constitutive and regulative norms. This seems very strong, but the proof system shows that this corresponds precisely to the aggregative cumulative transitivity rule for the combined system (Theorem 6). So if this rule is desired, then this semantics has to be chosen. For example, if we start from a system satisfying ACT, and then refining it with systems for constitutive and regulative norms, then we need to refine it in this way.

In this paper we also introduce new detailed logics for combining constitutive and regulative norms, deriving expressions x_p for x is obligatory because of the intermediate concepts p , or simply x meaning x is obligatory without referring to intermediate concept. We have extended each of the three above systems with a proof system for these refined expressions.

References

1. G. Boella and L. van der Torre. Regulative and constitutive norms in normative multi-agent systems. In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *Principles of Knowledge Representation and Reasoning*, pages 255–266, Whistler, Canada, 2004. AAAI Press.
2. G. Boella and L. van der Torre. A logical architecture of a normative system. In Lou Goble and John-Jules Ch. Meyer, editors, *Deontic Logic and Artificial Normative Systems*, volume 4048 of *Lecture Notes in Computer Science*, pages 24–35, Utrecht, The Netherlands, 2006. Springer.
3. C. Boutilier. Toward a logic for qualitative decision theory. In Jon Doyle, Erik Sandewall, and Pietro Torasso, editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pages 75–86, Bonn, Germany, 1994. Morgan Kaufmann.
4. J. Broome. *Rationality Through Reasoning*. Wiley-Blackwell, West Sussex, UK, 2013.
5. S. Danielsson. *Preference and Obligation: Studies in the Logic of Ethics*. Filosofiska freningen, Uppsala, 1968.

6. D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. van der Torre, editors. *Handbook of Deontic Logic and Normative Systems*. College Publications, London, 2013.
7. L. Goble. Murder most gentle: the paradox: deepens. *Philosophical Studies*, 64:217–227, 1991.
8. Guido Governatori and Antonino Rotolo. Bio logical agents: Norms, beliefs, intentions in defeasible logic. *Autonomous Agents and Multi-Agent Systems*, 17(1):36–69, August 2008.
9. D. Grossi and A. Jones. Constitutive norms and counts-as conditionals. In J. Horty, D. Gabbay, X. Parent, R. van der Meyden, and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*, pages 407–441. College Publications, London, 2013.
10. D. Grossi, J. J. Meyer, and F. Dignum. Counts-as: Classification or Constitution? An Answer Using Modal Logic. In Lou Goble and John-Jules Ch. Meyer, editors, *Deontic Logic and Artificial Normative Systems*, volume 4048 of *Lecture Notes in Computer Science*, pages 115–130. Springer, Berlin Heidelberg, 2006.
11. B Hansson. An analysis of some deontic logics. *Noûs*, pages 373–398, 1969.
12. A. Jones and M. Sergot. A formal characterisation of institutionalised power. *The Logic Journal of IGPL*, pages 427–443, 1996.
13. J. Lang and L. van der Torre. From belief change to preference change. In M. Ghallab, C.D. Spyropoulos, N. Fakotakis, and N. Avouris, editors, *Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 351–355, Amsterdam, 2008. IOS Press.
14. D.K. Lewis. *Counterfactuals*. Blackwell, Oxford, 1973.
15. L. Lindahl and J. Odelstad. Normative systems and their revision: An algebraic approach. *Artificial Intelligence and Law*, 11(2-3):81–104, 2003.
16. D. Makinson. On a fundamental problem in deontic logic. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems*, pages 29–54, Amsterdam, 1999. IOS Press.
17. D. Makinson and L. van der Torre. Input-output logics. *Journal of Philosophical Logic*, 29:383–408, 2000.
18. D. Makinson and L. van der Torre. Constraints for input/output logics. *Journal of Philosophical Logic*, 30(2):155–185, 2001.
19. D. Makinson and L. van der Torre. What is input/output logic? In B. Lowe, W. Malzkorn, and T. Rasch, editors, *Foundations of the Formal Sciences II: Applications of Mathematical Logic in Philosophy and Linguistics*, pages 163–174, 2003.
20. X. Parent and L. van der Torre. Aggregative deontic detachment for normative reasoning (short paper). In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, 2014.
21. X. Parent and L. van der Torre. “sing and dance!” input/output logics without weakening. In *12th International Conference on Deontic logic and Normative Systems (DEON 2014)*, 2014.
22. J. Rawls. Two concepts of rules. *The Philosophical Review*, 64(1):3–32, 1955.
23. A. Ross. Tû-tû. *Harvard Law Review*, 70:812–825, 1957.
24. J.R. Searle. *Speech Acts: an Essay in the Philosophy of Language*. Cambridge University Press, Cambridge (UK), 1969.
25. A. Stolpe. Normative consequence: The problem of keeping it whilst giving it up. In Ron van der Meyden and Leendert van der Torre, editors, *Proceedings of the 9th international conference on Deontic Logic in Computer Science, DEON '08*, pages 174–188, Berlin, Heidelberg, 2008. Springer-Verlag.
26. S. Tosatto, G. Boella, L. van der Torre, and S. Villata. Abstract normative systems: Semantics and proof theory. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *Principles of Knowledge Representation and Reasoning*, pages 358–368, Rome, 2012. AAAI Press.

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