Permissive and Regulative Norms in Deontic Logic

Maya Olszewski, Xavier Parent[*], Leendert van der Torre [†]

Abstract

This article provides a systematic analysis of the well-known notions of weak and strong permission in input/output (I/O) logic. We extend the account of permission initially put forward by Makinson and Van der Torre to the whole family of I/O systems developed during the last two decades. The main contribution is a series of characterisation results for strong permission, based on establishing the so-called non-repetition property. We also study an input/output logic not yet covered in the literature. It supports reasoning by cases—a natural feature of human reasoning. The output is not closed under logical entailment. At the same time, it avoids excess output using a consistency check—a technique familiar from non-monotonic logic. This makes it well suited for contrary-to-duty reasoning. The axiomatic characterisation is in terms of a generalised OR rule. We discuss the implications of all this for our understanding of the notion of the coherence of a normative system. Topics for future research are identified.[1]

1 Introduction

In the *Handbook of Deontic Logic and Normative Systems*, the chapter on permission [8] highlighted the existence of different varieties of permission. Makinson and Van der Torre [16] distinguished negative permission in the context of a set of regulative norms from static and dynamic permission in the context

[*]xavier@logic.at. Parent is supported by the Austrian Science Fund (FWF) [M-3240-N].

[†]leon.vandertorre@uni.lu. Van der Torre is financially supported by Luxembourg's National Research Fund (FNR) through the project Deontic Logic for Epistemic Rights (OPEN O20/14776480) and the (Horizon 2020 funded) CHIST-ERA grant CHIST-ERA19-XAI (G.A. INTER/CHIST/19/14589586), as well as by the European Union's Justice programme under grant agreement 101007420 (ADELE).

[1]This article is an extended version of a paper presented at the the 15th International Conference on Deontic Logic and Normative Systems (DEON) [18]. The main additions include extending the results to more logics, introducing an I/O operation left undefined in previous work (which we shall call here core basic output), and discussion of new topics (coherence, obligation under exception and deontic explanation). We thank an anonymous reviewer for his comments.

of permissive and regulative norms.[2] Boella and Van der Torre [2] introduced permission as an exception in the context of an explicitly hierarchical normative system that includes permissive norms. This was further developed by Hansen [6]. Moreover, Stolpe [32] studied permission as derogation.

All these notions of permission were studied using either variants of unconstrained input/output (I/O) logic as developed by Makinson and Van der Torre [14] or a version of constrained input/output logic based on so-called outfamilies [15] introduced by the same authors. However, over the past two decades, new variants of input/output logic have been introduced, in particular intuitionistic I/O logics [21, 19], I/O logics without weakening [20, 27], and I/O logics with a built-in consistency check [23, 26]. The first ones use intuitionistic logic rather than classical logic as the base logic. The second ones no longer require the output to be closed under logical entailment. The third ones have a consistency check which filters out excess output. They share some aspects with unconstrained input/output logic, e.g. they are monotonic. Moreover, they share some aspects with constrained input/output logic, e.g. they can handle contrary-to-duty reasoning. They have therefore been promoted as providing a good balance between the two traditional categories of input/output logic.

In this article, we provide a systematic analysis of weak and strong permissions from an input/output perspective. In particular, we address the following research questions:

1. How to define the notion of permission both semantically and proof-theoretically? A completeness or characterisation theorem is required.

2. How to generalise and apply traditional notions of permission in the input/output logic framework (as defined in [16]) to the whole family of existing I/O systems?

3. How to understand the notion of the coherence of a normative system on this basis?

4. How to define obligations with exceptions, that is, obligations detached from regulative norms that can be overridden by permissive norms?

To restrict the scope of this article, we shall focus on classical propositional logic and put aside the intuitionistic case. We shall also put aside the hierarchical structure of normative systems, prioritised norms, defeasible norms, dynamic permissions, formalisation of rights, and so-called outfamilies as developed in constrained input/output logic by Makinson and van der Torre [15]. All these topics are left for further research.

We will now explain how we intend to address our research questions. First of all, our analysis of permission is based on a generalised notion of coherence [7], combining insights from various studies in deontic logic. From the work

[2]Sometimes, permissive norms are seen as a subclass of regulative norms. In this article, when we refer to "regulative norms", we mean rules that detach obligations, and when we refer to "permissive norms", we mean rules that detach strong permissions.

on permission, we will adopt and generalise the notion of the cross-coherence of a normative system based on both permissive and regulative norms. From the work on constrained input/output logic, we will adopt the distinction between output constraint and input/output constraint, where the former is used to handle dilemmas, and the latter is used to handle contrary-to-duty reasoning. An example of a dilemma is: "It is permitted and forbidden to travel to Paris". An example of contrary-to-duty reasoning is: "You should not travel to Paris, but if you do, you are allowed to fly". In this article, we shall follow the terminology we used in [24], where we provided: an overview of the various kinds of reasoning involved in normative reasoning on matters such as dilemmas and contrary-to-duty scenarios, an overview of benchmark examples, and an overview of reasoning principles in normative reasoning.

Second, for our study we will look at so-called simple-minded output, basic output and reusable output of the traditional systems [14], I/O logics without weakening, and I/O logics with a built-in consistency check or constraint. So far, in the deontic logic literature, there are semantics for only two input/output logics with a built-in consistency constraint: semantics for so-called simple-minded output and semantics for reusable output (see Tables 1 and 3 later in this article). In particular, there is no semantics yet for basic output with a built-in consistency constraint. This means that reasoning by cases is not supported. Reasoning by cases is a natural feature of human reasoning. The statement to be proved is split into a finite number of cases, and each case is checked to see whether the proposition in question holds. We therefore define such an operator and provide a sound and complete axiomatisation using a generalised OR rule, deriving 'if a or b, then x or y' is obligatory $(a \vee b, x \vee y)$ from 'if a then x is obligatory'(a, x) and 'if b then o y is obligatory' (b, y).

In this article, "constrained" permission refers to a notion of permission with a built-in consistency check. To be more precise, for the semantics of constrained permission, we adapt the definitions of Makinson and Van der Torre [16], replacing their unconstrained input/output logic with the constrained version [26] and adapting their characterisations accordingly.

Using an idea proposed by Boella and Van der Torre [2], Hansen [6], Stolpe [32] and others that strong permission can be interpreted as an explicit exception, we define obligations with exceptions in a straightforward way: something is obligatory if it can be detached from regulative norms and if there is no strong permission to the contrary.

Finally, to explain why a normative system is or isn't coherent, we can choose the input/output logic, the notion of permission, the notion of consistency, whether to have quantification over contexts, and whether to use the output constraint or the input/output constraint. To explain why something is obligatory, we can not only provide the relevant norms together with a derivation of the obligation from the regulative norms, but we can also choose the input/output logic or the notion of consistency. It is even more challenging to explain why something is *not* obligatory. In standard deontic logic based on modal logic, such deontic explanations are provided by counter-models, but no such counter-model can be provided in norm-based deontic logics like input/output logic. We therefore illustrate how the different kinds of permissions can be used for such deontic explanations.

The article is structured as follows. Sections 2 and 3 lay the groundwork for subsequent sections. Section 2 gives an overview of all the existing I/O operations (for obligation). Our analysis of permission will presuppose them all. Section 3 introduces a new member to the I/O family. That is, it introduces an I/O operation left undefined in recent work [23, 26, 25], which we shall call "core basic". The I/O operation has three main features: reasoning by cases is supported, the output is not closed under logical consequence, and a built-in consistency check filters out excess output. Section 4 is the core of the article. We carry out a systematic analysis of weak and strong permissions with respect to the whole family of input/output logics presented in the previous sections. Characterisation results are given. Sections 5, 6 and 7 go one step further. Section 5 looks at the implications for our understanding of the notion of coherence, which has long been recognised as a fundamental formal property of a normative system.[3] Section 6 discusses the relationship between obligation and permission. Section 7 identifies topics for future research. Section 8 discusses related work. Section 9 concludes the article. Extra material is gathered in three appendices, including a proof of the non-repetition property (NRP) needed for the characterisation results for strong permission.

2 I/O logics for obligation: an overview

We shall start by giving an overview of the whole family of I/O logics for obligation, which our analysis of permission will presuppose.

Input/output logic is a general logical framework devised by Makinson and Van der Torre in order to reason about conditional norms [14, 15]. In input/output logic, the meaning of the deontic concepts is given in terms of a set of procedures yielding outputs from inputs. Thus, the semantics may be called "operational" rather than truth-functional. To some extent, the system can be viewed solely in terms of its input, output and transfer characteristics and without any knowledge of its internal workings, which remain "opaque" (a black box). Logic here is reduced to an ancillary role.

Let us call the normative system G_R the set of regulative conditional norms of the form (a, x), which means that "if a is the case, then it ought to be the case that x", where a is called the *body* and x the *head* of the norm. We call O an output operation, which for now the reader should understand as a black box which, given a context A and normative system G_R, outputs what ought to be the case as $O(G_R, A)$. Makinson and Van der Torre compare this to a secretarial assistant who has the task of "preparing inputs before they go into the machine, unpacking outputs as they emerge and, less obviously, coordinating

[3] The notion of the coherence of a normative system is the analogue of the notion of consistency in propositional logic. The former plays the same role in the realm of norms as the latter in the realm of facts.

the two" [14, p. 384]. This echoes the view in AI of logic as an added component to black boxes that are generated by machine learning algorithms [29].

In this section, we consider sixteen input/output logics, eight without a notion of coherence, and eight with a built-in notion of coherence. The coherence condition is represented by a consistency constraint. In the proof systems, the coherence condition leads to the presence of a consistency proviso restraining the application of some rules (aggregation and deontic detachment) as explained below.

2.1 Proof systems

The proof systems consist of sets of rules manipulating pairs of formulas. A derivation of (a, x) from a set of norms is viewed as a tree whose root is (a, x), whose leaves are elements of G_R, and each pair attached to a node is obtained from previous nodes using these rules. Different sets of rules give different systems. An overview of all the rules discussed in this article is provided in Appendix C. A more concise overview is provided in Figure 1. All the systems discussed in this article satisfy the requirements of strengthening the input (SI) and replacement of logical equivalents (EQ). This reflects the fact that we do not consider defeasible norms, prioritised norms, specificity, or syntactic differences between norms. Moreover, the systems are distinguished with respect to the following four dimensions:[4]

Aggregation The proof systems satisfy either restricted aggregation or unrestricted aggregation, represented as R-AND and AND respectively.[5]

Cases The proof systems do or do not satisfy reasoning by cases, a.k.a. the OR rule. Traditionally, those that do are called simple-minded and those that don't are called basic. In this article, we introduce a generalised form of reasoning by cases, ex-OR.

[4]In earlier work, a fifth dimension of throughput is also taken into consideration: input/output logics may or may not allow the input to be automatically carried out as output. Where it is allowed, the I/O operation is called "throughput". Throughput operations are less suited to normative reasoning because they validate the controversial rule of identity (ID)—from no premise, infer (y, y). They are thus less suitable for our analysis of permissions, and so we put them aside. Throughput operators are used for many other kinds of rule-based reasoning, e.g. in default logic. Technically, most concepts in this article can also be used for throughput operators simply by adding identity to all results. The traditional input/output logics of Makinson and Van der Torre consider three dimensions: cases, reusability and identity. The new dimensions are aggregation and closure. Closure is about the possibility of removing the property of closure from the obligation or permission operator under entailment, and aggregation is about the possibility of adding a consistency constraint to the semantics in order to filter out excess output.

[5]Even weaker systems that do not satisfy any aggregations have been studied by Ali Farjami [4], but without the ability to aggregate obligations, the role of logic seems very small. In [1], it is argued that aggregation is a minimal property of deontic logic. In [12] and [35], formalisations of reasoning in Sanskrit philosophy are presented. These logics do not contain aggregation, but they are deliberately weak. We call an input/output operation "loose" if there is no consistency constraint, and "extended" if closure under entailment holds.

Deontic detachment The proof systems satisfy the requirement of: no deontic detachment, its restricted version represented as restricted aggregative cumulative transitivity (R-ACT), or its unrestricted aggregative version represented as ACT. Deontic detachment has traditionally been assimilated with reusability and a supporting I/O system called reusable output.

Closure The proof systems either satisfy or do not satisfy closure under consequence, represented by the weakening of the output rule (WO).

In this article, we use classical propositional logic as the base logic $\mathcal{L}$ upon which the I/O system is defined. In addition to the terminology already introduced, for a set of norms G_R, then $h(G_R)$ is the set of all the heads of elements of G_R, and $b(G_R)$ is the set of all the bodies of elements of G_R. $G_R(A)$ is defined as $\{x : (a, x) \in G_R \ for\ some\ a \in A\}$. Cn represents the consequence operation from classical propositional logic. In order to facilitate some of the technical results that follow, we make the additional assumption that each (a, x) in G_R has a consistent fulfilment in the sense that the body and head are jointly consistent.

Definition 1 (Proof systems [14, 22, 26]). *Let G_R be a set of conditional norms, such that $\forall(a, x) \in G_R$, and (a, x) has a consistent fulfilment in the sense that $a \wedge x$ is consistent. We say that $(a, x) \in D_i^\nu(G_R)$ iff (a, x) is derivable from G_R using the rules of D_i^ν. We say that $(A, x) \in D_i^\nu(G_R)$ iff $(a, x) \in D_i^\nu(G_R)$ where a is a conjunction of formulas in A. Equivalently, we say that $x \in D_i^\nu(G_R, A)$.*

All considered proof systems are provided in Table 1. We use a unified notation for all systems: O_i^x/D_i^ν, where $i \in \{1, \ldots, 8\}$ and $\nu \in \{\llcorner\!\lrcorner, *\}$. The star superscript indicates systems without a built-in consistency constraint. All systems contain the basic rules SI and EQ, and some form of aggregation. Systems with an even subscript support a form of reasoning by cases, systems with the superscripts 3, 4, 7 and 8 satisfy some form of transitivity, and systems with subscripts 5–8 satisfy WO. The rules of each proof system are provided in Table 1 and the corresponding input/output logic is provided in Table 3. The original input/output logic, which was denoted as $out_1 - out_4$ in [14], has thus become $O_5^* - O_8^*$.

Table 2 shows the orderings/inclusions pertaining to the proof systems.

2.2 Semantics

Detachment (or *modus ponens*) is the core mechanism of the semantics. Existing semantics for the proof systems in Table 1 are listed in Table 3.

The semantics of simple-minded output O_5^*, traditionally called out_1, is $O_5^*(G_R, A) = Cn(G_R(Cn(A)))$. This semantics calculates the whole output in three steps. First, it takes the consequence set of input A, then it takes the image under G_R, and finally it takes the consequence set again. More complex semantics are based on this idea. For example, the semantics of basic output O_6^*, traditionally called out_2, is $O_6^*(G_R, A) = \cap\{Cn(G_R(Cn(V))) \mid A \subseteq V, V \text{ complete}\}$,

$$\frac{(a,x) \qquad (a,y)}{(a, x \wedge y)}\ \text{AND} \qquad \frac{(a,x) \qquad (a,y) \qquad a \wedge x \wedge y \nvdash \bot}{(a, x \wedge y)}\ \text{R-AND}$$

$$\frac{(a,x) \qquad (a \wedge x, y)}{(a, x \wedge y)}\ \text{ACT} \qquad \frac{(a,x) \qquad (a \wedge x, y) \qquad a \wedge x \wedge y \nvdash \bot}{(a, x \wedge y)}\ \text{R-ACT}$$

$$\frac{(a,x) \qquad (b,x)}{(a \vee b, x)}\ \text{OR} \qquad \frac{(a,x) \qquad (b,y)}{(a \vee b, x \vee y)}\ \text{ex-OR}$$

$$\frac{(a,x) \qquad x \vdash y}{(a,y)}\ \text{WO} \qquad \frac{(a,x) \qquad x \dashv\vdash y}{(a,y)}\ \text{EQ}$$

$$\frac{(a,x) \qquad b \vdash a}{(b,x)}\ \text{SI}$$

Figure 1: Input/output logic inference rules

where a complete set is either a maxi-consistent set, or the whole language $\mathcal{L}$. Each of the complete V represents one of the cases. We refer to Appendix A for the full definitions of these I/O operations. The input/output logics with a built-in consistency constraint, i.e. O_1 to O_8, are more involved than the traditional input/output logics O_1^* to O_8^*. The consistency constraint corresponds to a coherence condition. We introduce O_2 in the next section. It is new to the literature.

As the table illustrates, we do not yet have semantics for all proof systems. In particular, the semantics of O_5–O_7 have never been defined. The reason why these logics have not been studied is that they face so-called Van Fraassen's paradox (see [9, p. 79]). In systems O_5–O_8, we can derive Oq from Op and $O\neg p$. In other words, despite the built-in consistency constraint, we still have so-called deontic explosion: when a conflict of obligations arises, everything becomes obligatory. This problem has motivated the design of O_1–O_4.

The semantics can be used in various ways. For example, when we want to explain why a formula is not obligatory under O_5^* or O_6^*, we can compute the above sets of formulas and show that that formula is not part of any of them.

The proof-theoretical characterisation in the form of rules operating on pairs of formulas (as described in Table 1) is made possible by the following equivalences: $x \in D_i^\nu(G_R, \{a\})$ iff $(a,x) \in D_i^\nu(G_R)$, and $x \in O_i^x(G_R, \{a\})$ iff $(a,x) \in O_i^x(G_R)$. These equivalences have the status of a notational convention.

Table 1: Proof systems. x is a minimal set, + are derived rules. When marked with †, the OR rule is used instead of ex-OR.

System	SI	EQ	R-AND	AND	ex-OR/OR†	R-ACT	ACT	WO
D_1	x	x	x					
D_2	x	x	x		x			
D_3	x	x	+			x		
D_4	x	x	+		x	x		
D_5	x	+	x					x
D_6	x	+	x		x			x
D_7	x	+	+			x		x
D_8	x	+	+		x	x		x
D_1^*	x	x	+	x				
D_2^*	x	x	+	x	x†			
D_3^*	x	x	+	+		+	x	
D_4^*	x	x	+	+	x†	+	x	
D_5^*	x	+	+	x				x
D_6^*	x	+	+	x	x			x
D_7^*	x	+	+	+		+	x	x
D_8^*	x	+	+	+	x	+	x	x

Note finally that O_i^x is a closure operation:[6]

Reflexivity: $G_R \subseteq O_i^\nu(G_R)$

Monotonicity: $O_i^\nu(G_R) \subseteq O_i^\nu(G_R \cup H)$

Idempotence: $O_i^\nu(G_R) = O_i^\nu(O_i^\nu(G_R))$

3 A characterisation result for basic output with a consistency check

In this section, we study the basic I/O operation O_2. We define it, give a syntactical characterisation, and show completeness. This result is new to the literature.

Reasoning by cases is a rule that is often desirable to have and is intuitive. It can be particularly relevant for deontic explanation. Consistency constraints in input/output logic are required for contrary-to-duty reasoning, yet no input/output system combining consistency constraints and reasoning by cases has been proposed yet. For this reason, we introduce a new input/output logic that combines a form of reasoning by cases with consistency constraints.

This is relevant for permission because permission is defined in terms of obligation. Our primary goal in this article is to present a systematic analysis

[6]For further discussion, see the Deontic Logic Handbook chapter on input/output logic [21].

Table 2: The ordering/inclusions of the proof systems

D_1^ν	$\subseteq D_2^\nu$ $\subseteq D_3^\nu$	$\subseteq D_4^\nu$	for $\nu \in \{\textvisiblespace, *\}$
D_5^ν	$\subseteq D_6^\nu$ $\subseteq D_7^\nu$	$\subseteq D_8^\nu$	for $\nu \in \{\textvisiblespace, *\}$
D_i^ν	$\subseteq D_{i+4}^\nu$		for $i = 1, ..., 4$, $\nu \in \{\textvisiblespace, *\}$
D_i	$\subseteq D_i^*$		for $i = 1, ..., 8$

Table 3: Semantics introduced in the literature. "+" is introduced in this article. "−" indicates that the I/O operation has not been studied yet. An input/output operation is called "loose" if it validates WO and "core" if it does not validate WO. It is called "extended" if it has no built-in consistency constraint.

I/O operation	Name	References
O_1	*Core simple-minded output*	[26, 11]
O_2	*Core basic output*	+
O_3	*Core reusable simple-minded output*	[26, 11]
O_4	*Core reusable basic output*	−
O_5	*(Non-extended) simple-minded output*	−
O_6	*(Non-extended) basic output*	−
O_7	*(Non-extended) reusable simple-minded output*	−
O_8	*Non-extended) reusable basic output*	−
O_1^*	*Core extended simple-minded output*	[20]
O_2^*	*Core extended basic output*	[20]
O_3^*	*Core extended reusable simple-minded output*	[20]
O_4^*	*Core extended reusable basic output*	[28]
O_5^*	*Loose extended simple-minded output*	[14]
O_6^*	*Loose extended basic output*	[14]
O_7^*	*Loose extended reusable simple-minded output*	[14]
O_8^*	*Loose extended reusable basic output*	[14]

of weak and strong permissions with respect to the whole family of input/output logics (for obligation). This analysis would not be complete if all members were not covered.

Our new semantics is based on so-called single-step semantics, which we call core simple-minded output and refer to as O_1. It is defined as follows:

Definition 2 (Core simple-minded output [26])**.** *$x \in O_1(G, A)$ iff there exists some finite $M \subseteq G$ and a set $B \subseteq Cn(A)$ such that $M \neq \emptyset$, $B = b(M)$, $x \dashv\vdash \wedge h(M)$ and $\{x\} \cup B$ is consistent.*[7] *$O_1(G) = \{(A, x) : x \in O_1(G, A)\}$.*

In the proof system of O_2, the axiom ex-OR is added to allow reasoning by cases. It corresponds to the following semantics.

Definition 3 (Core basic output)**.** *$x \in O_2(G_R, A)$ iff $x \in O_1(G^\star_R, A)$ where $G^\star_R$ is the closure of G_R under ex-OR.*

$$\textit{ex-OR}\ \frac{(a, x) \qquad (b, y)}{(a \vee b, x \vee y)}$$

The following example shows what kinds of obligations can be inferred in O_2.

Example 1. Let s stand for snow, r for rain, g for good weather, d for driving and c for driving carefully. Then, using ex-OR, one derives $(\neg g, \neg d \vee c)$ from $(s, \neg d)$ and (r, c), assuming that $s \vee r \dashv\vdash \neg g$, i.e. we can derive that in bad weather we ought to drive carefully or not drive at all from the premises that we ought not to drive when it snows and that we ought to drive carefully when it rains.

The following phasing result will play a key role in the establishment of the characterisation theorem.

Lemma 1 (Phasing)**.** *$D_2(G_R) = D_1(G^\star_R)$, with $G^\star_R$ as in Definition 3.*

Proof. We show that any derivation in D_2 can be rewritten in such a way that ex-OR is applied first. This is because an application of R-AND (resp. SI) followed by ex-OR can be transformed into an application of ex-OR followed by R-AND (resp. SI). We treat EQ as a "silent rule" that may be applied anytime without an explicit justification.

- R-AND/ex-OR ⇒ ex-OR /R-AND

$$\frac{\text{R-AND}\ \dfrac{(b, y) \qquad (b, z)}{(b, y \wedge z)} \qquad (a, x)}{(a \vee b, x \vee (y \wedge z))}\ \text{ex-OR}$$

becomes

[7] $b(M)$ is the set of all the bodies of the rules in M, and $h(M)$ is the set of all the heads of the rules in M.

$$\dfrac{\dfrac{(a,x) \qquad (b,y)}{(a \vee b, x \vee y)}\text{ex-OR} \qquad \dfrac{(a,x) \qquad (b,z)}{(a \vee b, x \vee z)}\text{ex-OR}}{(a \vee b, x \vee (y \wedge z))}\text{R-AND}$$

because the consistency of $b \wedge y \wedge z$ entails $(a \vee b) \wedge (x \vee (y \wedge z)))$.

- SI/ex-OR $\Rightarrow$ ex-OR /SI

$$\dfrac{\dfrac{(a,x) \qquad b \vdash a}{(b,x)}\text{SI} \qquad (c,y)}{(b \vee c, x \vee y)}\text{ex-OR}$$

becomes

$$\dfrac{\dfrac{(a,x) \qquad (c,y)}{(a \vee c, x \vee y)}\text{ex-OR} \qquad b \vdash a}{(b \vee c, x \vee y)}\text{SI}$$

□

Theorem 1 (Soundness)**.**

$D_2(G_R, A) \subseteq O_2(G_R, A)$.

Proof. Let $x \in D_2(G_R, A)$. Hence $x \in D_2(G_R, a)$, where a is a conjunction of elements of A. So $x \in D_1(G_R^\star, a)$ [Lemma 1]. As a result, $x \in O_1(G_R^\star, a)$ [soundness for O_1] and $x \in O_1(G_R^\star, A)$ [monotonicity for O_1]. Hence $x \in O_2(G_R, A)$ [Definition 3]. □

It is worth noting that the argument for soundness, Theorem 1, does not follow the usual pattern of showing that the rules of the proof system are semantically valid. That can be verified independently.

Proposition 1. *O_2 validates the rules of D_2.*

Proof. R-AND. Assume $x \in O_2(G_R, a)$ and $y \in O_2(G_R, a)$ and that $a \wedge x \wedge y$ is consistent. In that case, $x \in O_1(G_R^\star, a)$ and $y \in O_1(G_R^\star, a)$. Since O_1 validates R-AND, $x \wedge y \in O_1(G_R^\star, a)$. By definition, $x \wedge y \in O_2(G_R, a)$.

SI. Assume $x \in O_2(G_R, a)$ and $b \vdash a$. So $x \in O_1(G_R^\star, a)$. Since O_1 validates SI, then $x \in O_2(G_R, b)$.

ex-OR. Let $x \in O_2(G_R, a)$ and $y \in O_2(G_R, b)$. Then $x \in O_1(G_R^\star, a)$ and $y \in O_1(G_R^\star, b)$ [def of O_2]. So $x \in D_1(G_R^\star, a)$ and $y \in D_1(G_R^\star, b)$ [completeness of D_1]. So $x \in D_2(G_R, a)$ and $y \in D_2(G_R, b)$ [Lemma 1]. So $x \vee y \in D_2(G_R, a \vee b)$ [ex-OR]. So $x \vee y \in D_1(G_R^\star, a \vee b)$ [Lemma 1]. So $x \vee y \in O_1(G_R^\star, a \vee b)$. So $x \vee y \in O_2(G_R, a \vee b)$.

□

Theorem 2 (Completeness)**.** $O_2(G_R) \subseteq D_2(G_R)$.

Proof. Assume $(A, x) \in O_2(G_R)$. By definition, $(A, x) \in O_1(G_R^\star)$. By completeness for core simple-minded I/O logic, $(A, x) \in D_1(G_R^\star)$. By Theorem 1, $(A, x) \in D_2(G_R)$ as required.

□

4 Permission

In this section, we present a systematic analysis of weak and strong permissions with respect to multiple input/output logics. In particular, we extend the work of Makinson and Van der Torre [16] and Olszewski et al. [18]. In the aforementioned articles, three kinds of permissions were distinguished: negative (weak) permissions, positive static (strong) permissions and positive dynamic permissions. However, in our article, we focus only on the first two kinds. Dynamic permission situates itself behaviourally in between weak and strong permission and is of much interest; however axiomatising dynamic permission is a challenging endeavour and the methods proposed in [16] no longer work for logics with a built-in consistency constraint [18].

Weak permission is defined as what is not prohibited, and strong permission is what follows from regulative norms combined with an explicit permissive norm. We first provide an analysis of weak permission, and then an analysis of strong permission.

4.1 Weak permission

The articles mentioned above analysed permission operations for six different underlying input/output logics in total, O_5^*–O_8^* as well as O_1 and O_3. They presented a mechanism for transitioning from the rules of the axiomatic characterisation of the obligation operation to the rules of the axiomatic characterisation of weak permission (called *inverse rules*) and strong permission (called *subverse rules*). Furthermore, they showed soundness and completeness for strong permission rules.

Completeness was not proven for weak permission. This is because weak permission is not a normative notion per se, in the sense that it does not come from the presence of norms, but rather from their absence. Hence, when we base weak permission on obligation, we in fact have to deal with underivability, a property of rule-based systems that is much more difficult than its preference-based counterpart, which is the counter-model.

The notion of weak permission dates back to the very start of modern deontic logic, when von Wright used permission as a primitive and defined the other notions from that primitive. He stated that whatever is not permitted is forbidden, and that whatever is such that its negation is not permitted is obligatory [36]. It is now more common to take the obligation operation as the primitive and define permission in terms of it. Still, the basic idea is the same.

Definition 4. *Let O_i^ν be an output operation, and let G_R be a set of regulative norms and A the context. Then x is said to be weakly permitted given context A and normative code G_R iff x is not prohibited given the same context and code:*

$$x \in WP_i^\nu(G_R, A) \textit{ iff } \neg x \notin O_i^\nu G_R, A)$$

A definition like this has certain implications. Since the obligation operation can be characterised by a set of rules, this dictates the way that permission behaves when defined in terms of obligation. For instance, since weak permission is defined negatively, i.e. something is permitted iff it is not prohibited, this entails that if the underlying logic is such that it allows the derivation of many obligations, there will be fewer things permitted. Similarly, if the input/output logic does not allow many obligations to be derived, the number of permissions will be higher.

To see the effect of the semantic definition of weak permission in terms of the obligation operation, we look at the general forms of the obligation rules and show how they change into permission rules. The rules of D_i^ν have the following form:

$$\begin{aligned}&\text{(HR): } (\alpha_j, \varphi_j) \in O_i^\nu(G_R)\ (j \leq n)\ \&\ \theta_k \in Cn(\gamma_k)\ (k \leq m)\\ &\quad \&\ \bigwedge_{l=0}^{n'} (\alpha_l \wedge \varphi_l) \nvdash \bot\ \Rightarrow (\beta, \psi) \in O_i^\nu(G_R)\end{aligned}$$

Their *inverses* have the form:

$$\begin{aligned}&\text{(HR)}^{-1}\text{: } (\alpha_j, \varphi_j) \in O_i^\nu(G_R)\ (j < n)\ \&\ (\beta, \neg\psi) \in WP_i^\nu(G_R)\\ &\quad \&\ \theta_k \in Cn(\gamma_k)\ (k \leq m)\ \&\ \bigwedge_{l=0}^{n'} (\alpha_l \wedge \varphi_l) \nvdash \bot\\ &\quad \Rightarrow (\alpha_n, \neg\varphi_n) \in WP_i^\nu(G_R)\end{aligned}$$

Proposition 2. *If an output operation O_i^ν satisfies the (HR) rule, then the corresponding weak permission WP_i^ν satisfies the inverse rule (HR)$^{-1}$.*

Proof. Let G_R be a set of norms and assume the following:

1. O_i^ν satisfies (HR)
2. $(\alpha_i, \varphi_i) \in O_i^\nu(G_R)$ for $i < n$
3. $(\beta, \neg\psi) \in WP_i^\nu(G_R)$
4. $\theta_j \in Cn(\gamma_j)$
5. $\bigwedge_{k=0}^{n'} (\alpha_k \wedge \varphi_k) \nvdash \bot$

By 3. and the definition of weak permission, we have that:

6. $(\beta, \psi) \notin O_i^\nu(G_R)$

By 2., 4. and 5., we know that all premises of the (HR) rule hold, with the exception of one premise, namely that of $(\alpha_n, \varphi_n) \in O_i^\nu(G_R)$, which is unknown. By 3., we know that the conclusion of (HR) does not hold. By 1., we know that O_i^ν satisfies (HR). Hence, the only premise with an unknown status cannot hold, thus $(\alpha_n, \varphi_n) \notin O_i^\nu(G_R)$. By the definition of weak permission, we get $(\alpha_n, \varphi_n) \in WP_i^\nu(G_R)$. □

4.2 Strong permission

Apart from negative permission, the other widely acknowledged kind of permission is strong permission, also referred to as positive or explicit permission. Strong permissions are permissions that have been explicitly granted, or that follow from explicitly granted permissions and obligations. We define them in the following way:

Definition 5. *Let O_i^ν be an output operation, G_R a set of regulative norms, G_P a set of permissive norms and A the context. Then x is said to be strongly permitted given context A and normative codes G_R and G_P iff x is output from the regulative code together with a single permissive norm from the same context:*

$$x \in SP_i^\nu G_R, G_P, A) \text{ iff } x \in O_i^\nu(G_R \cup Q, A)$$

where $Q \subseteq G_P$ is a singleton or an empty set.

The above definition gives a general pattern, which may be instantiated using any I/O operation O_i^ν as you think fit. One reason for requiring that Q should be a singleton is that G_P may contain two pairs whose heads contradict one another. For instance, it may be permitted to open and close the window at the same time. This is usually called a bilateral permission. If O_i^ν is closed under entailment, everything is permitted, which may be considered counter-intuitive. This is because of the principle *"ex falso sequitur quodlibet"*.

Following von Wright [37], it is common to define strong permission in the above way, not allowing permissive norms to aggregate. The classical example is the drinking-and-driving case. It is plausible to say that the permissions "Alice is allowed to drink" and "Alice is allowed to drive" both hold at the same time without having "Alice is allowed to drink and drive" hold.

The transformation of the Horn rule that corresponds to strong permission is called the *subverse* rule, and is defined in the following way:

$$\begin{aligned}
(\mathrm{HR})^{\downarrow}\text{: } & (\alpha_j, \varphi_j) \in O_i^\nu(G_R)\ (j < n)\ \&\ (\alpha_n, \varphi_n) \in SP_i^\nu(G_P, G_R) \\
& \&\ \theta_k \in Cn(\gamma_k)\ (k \leq m)\ \&\ \bigwedge_{l=0}^{n'} (\alpha_l \wedge \varphi_l) \nvdash \bot \\
& \Rightarrow (\beta, \psi) \in SP_i^\nu(G_P, G_R)
\end{aligned}$$

We illustrate the notion of a subverse rule with an example. A pair with the superscript o is an obligation. A pair with the superscript p is a permission.

Example 2 (Voting)**.** Suppose that the obligation operation satisfies restricted aggregation (R-AND). Let lux stand for being a Luxembourgish citizen, pl for being a Polish citizen, and v_{lux} and v_{pl} for voting as a Luxembourgish or Polish citizen in the EU elections. Seeing that in Luxembourg there is an obligation to vote, and in Poland there is not, and every EU citizen is only allowed to vote in the EU elections once, we have that $(lux \wedge pl, v_{lux})^o$ and $(lux \wedge pl, v_{pl})^p$, but not $(lux \wedge pl, v_{lux} \wedge v_{pl})^p$, since $lux \wedge pl \wedge v_{lux} \wedge v_{pl}$ is inconsistent. In a case where the obligation operation satisfies the unrestricted version of aggregation (AND), we are able to derive $(lux \wedge pl, v_{lux} \wedge v_{pl})^o$. This is counter-intuitive.

Proposition 3. *If an output operation O_i^ν satisfies the (HR) rule, then the corresponding strong permission SP_i^ν satisfies the subverse rule (HR)$^\downarrow$*

Proof. Let G_R be a set of norms and assume the following:

1. O_i^ν satisfies (HR)
2. $(\alpha_i, \varphi_i) \in O_i^\nu(G_R)$ for $i < n$
3. $(\alpha_n, \varphi_n) \in SP_i^\nu(G_R)$
4. $\theta_j \in Cn(\gamma_j)$
5. $\bigwedge_{k=0}^{n'}(\alpha_k \wedge \varphi_k) \nvdash \bot$

By 3. and the definition of strong permission, we have that:

6. $(\alpha_n, \varphi_n) \in O_i^\nu(G_R \cup Q)$ where $Q \subseteq G_P$ singleton or empty.

By 1., we know that O_i^ν satisfies (HR). Since the O_i^ν we consider here are monotonic, we have from 2. that also:

7. $(\alpha_i, \varphi_i) \in O_i^\nu(G_R \cup Q)$ for $i < n$.

Let $H = G_R \cup Q$. By 4., 5., 6., and 7., we know that all the premises of the (HR) rule hold for normative system H. By 1., O_i^ν satisfies (HR), hence the conclusion of (HR) also holds, and we get $(\beta, \psi) \in O_i^\nu(H)$. Since $H = G_R \cup Q$ with $Q \subseteq G_P$, whether singleton or empty, we get by our definition of strong permission that $(\beta, \psi) \in SP_i^\nu(G_R)$. □

Now that we know which rules strong permission satisfies with respect to the base logic, we want to discuss which rules can be desirable or problematic when reasoning about permission.

First, let's consider the case where the content of a permission is inseparable. A study of inseparability was conducted in [6]. We illustrate inseparability with the following example:

Example 3. This is a version of Feldman's medical example from [5, p. 87]. Let i stand for a patient having an illness, with m_1 and m_2 being two medicines used for the treatment of this illness, such that m_1 is only safe to use in combination with m_2. Then $(i, m_1 \wedge m_2)^p$ holds, but not $(i, m_1)^p$.

Where the contents of permissions are inseparable, weakening of the output is not recommended.

Second, we argue that consistency constraints are beneficial, as Example 2 showed. They also prevent the pragmatic oddity [30]. Pragmatic oddity arises when it is possible to derive from an obligation to keep one's promise and an obligation to apologise if one does not keep it that one should keep one's promise and apologise for not keeping it. For an in-depth analysis of pragmatic oddity in input/output logic, see [23]. Here we consider a permission variant of it:

Example 4 (Broken promise)**.** Let p stand for keeping a promise and e for explaining why the promise was not kept. Then, from assumptions that one should keep one's promise $(\top, p)^o$ and that if one does not keep it one is allowed to explain the reason why $(\neg p, e)^p$, it is possible to derive that one is permitted to keep one's promise and explain why one did not keep it $(\neg p, p \wedge e)^p$. Such a derivation goes through standard aggregation (systems O_5-O_8, O_1^*-O_2^* and O_5^*-O_8^*). But it is blocked by the consistency proviso of the restricted version (systems O_1-O_4).

4.3 Characterisation result (strong permission)

Proposition 3 corresponds to the soundness of the system and is straightforward. Completeness, however, is quite particular. Makinson and Van der Torre have shown that completeness amounts to having the non-repetition property for any derivation in the given system [16]. This is the property:

> *Non-repetition property* (NRP). A derivable pair (a, x) can always be derived in such a manner that every leaf node (in the derivation) is used only once.

NRP has been used in other contexts, for example to capture the idea that every premise is a resource that can be used only once. In the proof theory literature, it is known as contraction closure [13].

To see why NRP is sufficient for completeness of the subverse rules, recall the semantic definition of strong permission. We say that $(a, x) \in SP_i^\nu(G_P, G_R)$ iff $(a, x) \in O_i^\nu(G_R \cup Q)$ for $Q \subseteq G_P$, whether it is a singleton or empty. The main idea of strong permission is that we are only allowed to use one permissive norm at a time. This is also reflected in the subverse rules $(D_i^\nu)^\downarrow$, where one premise at most is allowed to be a permissive premise, and if that is the case, then the conclusion is a permissive norm, which may be used as a premise for further rule applications. However, if a permissive leaf is used twice, the derivation necessarily has two sub-derivations, each of which results in permissive conclusions that end up meeting in some rule. However, two permissive norms may never be used together. Recall the drinking-and-driving example. The two permissive norms permit drinking and driving separately, but they are not allowed to "join" into permitting drinking and driving. Hence, if we are looking at SP_i^ν, then the NRP of D_i^ν is a sufficient condition for completeness, since every formula from $O_i^\nu(G_R \cup Q, A)$ will be derivable in D_i^ν from $G_R \cup Q$ using leaf Q once at the most . The rest follows from the completeness of O_i^ν with regard to D_i^ν.

Proposition 4. *D_1^*, D_3^*, D_5^*, D_6^* D_7^*, D_1 and D_3 satisfy the requirements of NRP.*

To ease readability, the full proof is given in the Appendix. Here, we illustrate the core of the technique. It is based on phasing, which is rewriting proofs, or part of proofs, with rules being applied in a specific order.

For D_3^*, applications of R-ACT followed by SI (on the left-hand side below) must be rewritten into applications of SI followed by R-ACT (on the right-hand side below). However, the consistency constraint blocks the permutation, because the fact that $a \wedge x \wedge y$ is consistent does not imply that $a \wedge b \wedge x \wedge y$ is consistent:

$$\text{SI}\,\dfrac{\text{R-ACT}\,\dfrac{(a,x) \qquad (a \wedge x, y)}{(a, x \wedge y)}}{(a \wedge b, x \wedge y)} \qquad\qquad \dfrac{\dfrac{(a,x)}{(a \wedge b, x)}\,\text{SI} \qquad \dfrac{(a \wedge x, y)}{(a \wedge b \wedge x, y)}\,\text{SI}}{(a \wedge b, x \wedge y)}\,\text{R-ACT}$$

$$\text{if } a, x, y \nvdash \bot \qquad\qquad\qquad\qquad \text{if } a, b, x, y \nvdash \bot$$

To overcome this problem, we consider only subparts of derivations *above* the R-ACT rule, and we show that they can be phased in the way we need them to be.

For D_2^* and D_2, the problem cannot be overcome so easily. This is why they are not mentioned in the statement of the theorem.

Proposition 4 says nothing about D_2 and D_2^*, and for good reason. We find it rather unlikely that NRP holds for them. This does not mean that SP_2 and SP_2^* are non-axiomatisable.

For D_2, consider the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{(a,x) \qquad (b,y)}{(a \vee b, x \vee y)}\,\text{ex-OR}}{((a \vee b) \wedge (a \vee c), x \vee y)}\,\text{SI} \qquad \dfrac{\dfrac{(a,x) \qquad (c,z)}{(a \vee c, x \vee z)}\,\text{ex-OR}}{((a \vee b) \wedge (a \vee c), x \vee z)}\,\text{SI}}{((a \vee b) \wedge (a \vee c), (x \vee y) \wedge (x \vee z))}\,\text{R-AND}}{(a \vee (b \wedge c), x \vee (y \wedge z))}\,\text{EQ}$$

Note that the consistency constraint applied to R-AND is the following: $(a \vee b) \wedge (a \vee c) \wedge (x \vee y) \wedge (x \vee z) \nvdash \bot$. The only derivation that would allow us to derive the same conclusion from each premise used only once, given the rules of D_2, is the following:

$$\dfrac{\dfrac{\dfrac{(b,y)}{(b \wedge c, y)}\,\text{SI} \qquad \dfrac{(c,z)}{(b \wedge c, z)}\,\text{SI}}{(b \wedge c, y \wedge z)}\,\text{R-AND (dashed)} \qquad (a,x)}{(a \vee (b \wedge c), x \vee (y \wedge z))}\,\text{ex-OR}$$

The dashed line means the derivation is blocked. There is no guarantee that R-AND can be applied in this second derivation as it requires that $b \wedge c \wedge y \wedge z \nvdash \bot$, which is not entailed by the much weaker requirement of the first derivation. So, if (a,x) happens to be a permissive norm, the subverse rules of $(D_2)^{\downarrow}$ do not allow us to derive $(a \vee (b \wedge c), x \vee (y \wedge z))$ from regulative premises $\{(b,y),(c,z)\}$ and permissive premise (a,x) (due to the requirement that one premise in the derivation at most can be a permissive norm). However, we are well in the situation that $(a \vee (b \wedge c), x \vee (y \wedge z)) \in O_2(\{(a,x),(b,y),(c,z)\})$, hence $(a \vee (b \wedge c), x \vee (y \wedge z)) \in SP_2(\{(b,y),(c,z)\},\{(a,x)\})$.

What creates the issue is the combination of the consistency constraint and the weakening that happens through the ex-OR rule. If ex-OR is applied first, the consistency constraint is on the weakened elements, and it is no longer possible to invert ex-OR and R-AND in order to get an alternative derivation.

For D_2^*, this is the same. Consider:

$$\dfrac{\dfrac{\dfrac{(a, x \wedge y) \qquad (b, x \wedge z)}{(a \wedge b, x \wedge y \wedge z)}\text{SI+AND} \qquad \dfrac{(a, x \wedge y) \qquad (c, y \wedge z)}{(a \wedge c, x \wedge y \wedge z)}\text{SI+AND}}{((a \wedge b) \vee (a \wedge c), x \wedge y \wedge z)}\text{OR}}{(a \wedge (b \vee c), x \wedge y \wedge z)}\text{EQ}$$

There is no derivation that uses each leaf only once.This shows that the establishment of NRP is not as trivial as it may first seem.

As a corollary result, one gets:

Corollary 1 (Completeness)**.** *The subverse rulesets are enough to fully characterise the corresponding strong permission when the underlying I/O operation is one of the following: O_1^*, O_3^*, O_5^*, O_6^* O_7^*, O_1 and O_3, i.e. $(D_1^*)^{\downarrow}$ (respectively $(D_3^*)^{\downarrow}$, $(D_5^*)^{\downarrow}$, $(D_6^*)^{\downarrow}$ $(D_7^*)^{\downarrow}$, $(D_1)^{\downarrow}$ and $(D_3)^{\downarrow}$) are sufficient to fully characterise SP_1^* (respectively SP_3^*, SP_5^*, SP_6^* SP_7^*, SP_1 and SP_3).*

5 Coherence

It is often assumed that norms do not have truth values. This is known as Jorgensen's dilemma [10]. It is thus also often concluded that we cannot talk about the consistency or inconsistency of a normative system. However, it is generally useful to have a formal notion that reflects the property of a normative system that is intuitively well-behaved, compatible or conflict-free, similar to the way that consistency for propositions reflects a property that the description is conflict-free. This property for normative systems we shall call *coherence*.

As a starting point, most rule-based systems may be called coherent if, under some context, they do not prescribe contradictory actions. Hence, a rule-based system that is coherent is unproblematic in the sense that it can be applied without further interpretation or discussion. A rule-based system that is incoherent needs to be handled more carefully; it may require interpretation or additional deliberation.[8]

In deontic logic, this is not the only possible definition. We could also say that a violation is a kind of incoherence, and that contrary-to-duty reasoning

[8]In this article, incoherence is understood as a situation where dilemmas exist or may occur. Clearly, other definitions of incoherence could be given as well. For example, assuming moral dilemmas are ubiquitous in the real world, some people may be hesitant to call such a rule-based system incoherent. As this seems to refer more to a linguistic interpretation of "coherence" than to logical analysis, we do not discuss this point further in this article.

is an attempt to make a normative system coherent again. For this reason, formalising coherence is particularly challenging for normative systems.

In general, the coherence of a normative system can take many forms. In this section, we discuss various ways of conceptualising coherence. We do not go into the reasons why a normative system is coherent or incoherent[9] or whether we ought to strive toward coherence.[10]

5.1 The many faces of coherence

Even if the coherence of a normative system cannot be defined as the consistency of a set of norms in a particular context, it can be defined in terms of the consistency of the *detachments* of the norms in that context. For example, even if the norm "There should be world peace" cannot be given a truth value, the proposition "There is world peace" is either true or false. In other words, once we define a way to apply the norms, then we can define the different notions of coherence in terms of the consistency of expressions such as "There is world peace". In this article, we use an abstract operator O, as studied in the input/output logic framework, to apply norms in a specific context in this way.

5.1.1 Obligation coherence

This section refers to obligation coherence (we do not yet consider permissions).

We distinguish between two notions related to the coherence of normative systems that appear in the literature on input/output logic, namely *output consistency* and *input/output consistency*. These have been discussed by Makinson and Van der Torre on the topic of constrained input/output logic [15].

First, we can require coherence in the normative system. Output consistency requires the absence of rules with an inconsistent head, i.e. $(a, \bot) \notin G_R$ for any a. The input/output consistency of a normative system requires that for every norm in the normative system, the head is consistent with the body, i.e. $\forall(a, x) \in G_R,\ a \land x$ are consistent. We say that such a norm has a *consistent fulfilment*. In this article, we assume that all norms have a consistent fulfilment.[11]

[9] Incoherence in normative systems can either come from design errors or can be intentional in the sense that the designer wants to point the user to a dilemma that needs further reflection. For example, norms being heuristic rules of thumb, parts of the system to be regulated can be unknown or, as in legal texts, the system may also regulate unknown future scenarios [3].

[10] In other words, we do not assume that there is a norm stating that it is forbidden to make a normative system incoherent. There are authors who disagree with this claim. For example, Ruth Barcan Markus argues that the presence of (moral) dilemmas is not an indication of inconsistency in a system of norms [17]. However, she has a very specific notion of inconsistency in mind and, as we shall elaborate below, coherence can take many forms. A normative system may be coherent according to one definition, and incoherent according to another.

[11] This is done for practical reasons: we do not have to consider borderline cases in the proofs. The generalisation to the general case where norms do not have to have a consistent fulfilment is straightforward as such borderline cases need to be defined explicitly.

Secondly, we can require stronger versions of output consistency and input/output consistency that require consistency not only in the elements of the normative system but also in everything detached from them. For a context A, output consistency is defined as $O(G_R, A)$ being consistent, whereas input/output consistency is defined as $O(G_R, A) \cup \{A\}$ being consistent. Output consistency can be used to handle conflicts in normative systems, while input/output consistency can be used to handle contrary-to-duty (CTD) reasoning in the sense that the consistency constraint on the context A and on the output limits the norms to be considered to those that are applicable to the violation context and not the general norms that are applicable when there is no violation. Note that input/output consistency implies output consistency. Furthermore, if $A \subseteq O(G_R, A)$, i.e. O is a throughput operator (satisfying the identity rule ID — see Section 2), then the two notions collapse to a single notion. The coherence of a normative system can be defined in terms of output consistency or input/output consistency.

One can think of imposing either one of the two notions of consistency to the pair (G_R, A) at different levels. First, one can require absolute consistency, where $O(G_R, A)$ needs to be consistent. Next, one can refine this notion either by context and some quantification thereof or by a generator set. In the former case, one can require consistency only for some contexts, or for all contexts. In the latter case, we can ask for consistency for only those contexts that are bodies of explicit norms in G_R, which means that they correspond to situations foreseen by the source of the normative code.

Definition 6 formalises these various kinds of obligation coherence. With only regulative norms G_R and context A, coherence depends on four elements. The first element is an operation O associating a set of obligations with G_R and A.[12] The second element is the choice between output consistency and input/output consistency. The third element is quantification over relevant contexts. The fourth element is the notion of consistency for the base language.[13]

Definition 6 (Obligation coherence)**.** *Let $\mathcal{L}$ be a logical language, G_R a set of pairs of $\mathcal{L}$ having a consistent fulfilment, A a subset of $\mathcal{L}$, and O a function that associates subsets of $\mathcal{L}$ with such G_R and A. Moreover, let cons be a Boolean function telling us whether a subset of $\mathcal{L}$ is consistent or not. The pair (G_R, A) is:*

O$\top$ coherent *iff* $cons(O(G_R, A))$

O$\exists$ coherent *iff* $\exists B \subseteq \mathcal{L} : cons(O(G_R, A \cup B))$

[12] The same normative system G_R can be coherent in one logic and incoherent in another one. For example, consider the normative system $G_R = \{(a, x), (a, \neg x)\}$ in context $A = \{a\}$. Then, in a logic that allows for aggregation in the consequent of the norms, G_R is incoherent in nearly all the considered definitions. However, in a logic that does not allow for any kind of aggregation or strengthening of the consequent, then G_R appears to be coherent. Hence, in the same way that logics need to be handpicked for the application domain, the right coherence needs to be adapted to both the logic and the application.

[13] For example, when the base language is propositional logic, we can distinguish direct inconsistency (we have two sentences ϕ and $\neg\phi$ in the set) from indirect inconsistency (we can derive a contradiction from the set in propositional logic).

O∀ coherent *iff* $\forall B \subseteq \mathcal{L} : cons(A \cup B) \Rightarrow cons(O(G_R, A \cup B))$

OR coherent *iff* $\forall (a,x) \in G_R : cons(O(G_R, A \cup \{a\}))$

IO⊤ coherent *iff* $cons(A \cup O(G_R, A))$

IO∃ coherent *iff* $\exists B \subseteq \mathcal{L} : cons(A \cup B \cup O(G_R, A \cup B))$

IO∀ coherent *iff* $\forall B \subseteq \mathcal{L} : cons(A \cup B) \Rightarrow cons(A \cup B \cup O(G_R, A \cup B))$

IOR coherent *iff* $\forall (a,x) \in G_R : cons(A \cup \{a\} \cup O(G_R, A \cup \{a\}))$

We use the Boolean function coh to denote whether a pair (G_R, A) *is coherent. We use* X *as a variable ranging over* $\{I, IO\}$ *and we use* Y *as a variable ranging over* $\{\top, \exists, \forall, R\}$*. If* (G_R, A) *is XY-coherent, we denote this as* $coh_X^Y(G_R, A)$*. For instance, if* (G_R, A) *is IO∃-coherent, we denote this as* $coh_{IO}^{\exists}(G_R, A)$*.*

Note that if $coh_X^{\forall}(G_R, A)$, then also $coh_X^{\top}(G_R, A)$ and $coh_X^R(G_R, A)$. Furthermore, if $coh_{IO}^Y(G_R, A)$, then also $coh_O^Y(G_R, A)$. Moreover, if O is a throughput operator, then $coh_O^Y(G_R, A)$ iff $coh_{IO}^Y(G_R, A)$.

To illustrate how the different notions of coherence from Definition 6 behave on the same normative system, consider the following example of a potential dilemma.

Example 5 (Potential conflict between obligations)**.** Let $G_R = \{(a,x), (b,\neg x)\}$ and let O be such that[14] $O(G_R, \{x\}) = \emptyset$, $O(G_R, \{x,a\}) = \{x\}$, $O(G_R, \{x,b\}) = \{\neg x\}$ and $O(G_R, \{x,a,b\}) = \{x \wedge \neg x\}$, $cons(\{a,b,x\})$ and its subsets, and not $cons(\{x \wedge \neg x\})$ and its supersets. Then it is the case that $coh_O^{\top}(G_R, \{x\})$, $coh_O^{\exists}(G_R, \{x\})$ and $coh_O^R(G_R, \{x\})$, but $\neg coh_O^{\forall}(G_R, \{x\})$, since $cons(\{a,b,x\})$ but $\neg cons(O(G_R, \{a,b,x\}))$. Furthermore, we have $coh_{IO}^{\top}(G_R, \{x\})$ and $coh_{IO}^{\exists}(G_R, \{x\})$, but $\neg coh_{IO}^R(G_R, \{x\})$, since $\neg cons(\{x,b\} \cup O(G_R, \{x,b\}))$. For the same reason, $\neg coh_{IO}^{\forall}(G_R, \{x\})$.

Proposition 5. *Once a normative system is incoherent, we cannot make it coherent by adding new regulative norms.*

$$coh_X^Y(G_R \cup G, A) \rightarrow coh_X^Y(G_R, A) \text{ with respect to } O_i^x$$

Equivalently,

$$\neg coh_X^Y(G_R, A) \rightarrow \neg coh_X^Y(G_R \cup G, A)$$

Proof. Assume $coh_X^Y(G_R \cup G, A)$ for some G_R, G normative sets and context A, as well as some notion of obligation coherence, and let O_i^x be any of the input/output logics presented in this section. Since O_i^x is monotonic, it is the case that $O_i^x(G_R \cup G) \supseteq O_i^x(G_R)$. Hence, if $cons(O_i^x(G_R \cup G))$, then also $cons(O_i^x(G_R))$ (since we use propositional logic for $\mathcal{L}$, we use the standard notion of consistency). Plugging this into the definition of coherence gives us the result. We will just show it for one of the possibilities here. Assume $coh_O^{\forall}(G_R \cup G, A)$. Then $\forall B : cons(A \cup B) \Rightarrow cons(O(G_R \cup G, A \cup B))$. Hence, $\forall B : cons(A \cup B) \Rightarrow cons(O(G_R, A \cup B))$, and so $coh_O^{\forall}(G_R, A)$. □

[14]This holds for any input/output operation defined in this paper.

5.1.2 Permission coherence

So far, we have discussed the notion of coherence in the context of only a single set of regulative norms. But what happens if we have two sets of norms, one regulative and one permissive, and want to ensure coherence between those sets? Makinson and Van der Torre proposed a different notion of coherence that takes into account both of the above-mentioned sets of norms and defines the coherence between them, which they call *cross-coherence* [16]. The idea is that a set of regulative norms G_R is cross-coherent with a set of permissive norms G_P if it can never be the case that, under some context, something is obligatory while its opposite is strongly permitted. If we assume that everything that is obligatory is also permitted, then it follows that the incoherence of the obligation set implies cross-incoherence between the regulative and permissive sets. In other words, the coherence of the obligation set is a condition for the coherence of the permissive set, albeit not the only one. Hence, this notion should possibly be considered in conjunction with some notion of coherence on the obligation set in order to provide the most information.

The above notion is based on von Wright's understanding of the compatibility between an obligation set and a permission set, where "a mixed set of norms is consistent, its members compatible if, and only if, each one of the members of its P-part is, individually, compatible with its 0-part." [37, p. 144][15]

With both regulative and permissive norms G_R, G_P and context A, coherence depends on five elements and can be thought of as extending obligation coherence from Definition 6 to include permissive norms. As before, it depends on O, the choice between output consistency and input/output consistency, the quantification over relevant contexts, and *cons*. In addition, it depends on a P function associating a set of permissions with G_R, G_P and A. Since permissions can be conflicting (e.g. it is permitted to open the window while at the same time it is permitted to close the window), we only demand that each individual permission is consistent with the set of obligations.

Definition 7 (Coherence)**.** *Let $\mathcal{L}$, G_R, A, O and cons be as before. Moreover, let G_P be a set of pairs of $\mathcal{L}$ having a consistent fulfilment (like G_R), and let P be a function that associates subsets of $\mathcal{L}$ with such G_R, G_P and A (analogous to O). The triple (G_R, G_P, A) is:*

O⊤ coherent *iff* $\forall p \in P(G_R, G_P, A)$: $cons(\{p\} \cup O(G_R, A))$

O∃ coherent *iff* $\exists B: \forall p \in P(G_R, G_P, A \cup B)$: $cons(\{p\} \cup O(G_R, A \cup B))$

O∀ coherent *iff* $\forall B: cons(A \cup B) \Rightarrow (\forall p \in P(G_R, G_P, A \cup B)$: $cons(\{p\} \cup O(G_R, A \cup B)))$

OR coherent *iff* $\forall (a, x) \in G_R \cup G_P: \forall p \in P(G_R, G_P, A \cup \{a\})$: $cons(\{p\} \cup O(G_R, A \cup \{a\}))$

[15] Hansen challenges von Wright's idea of coherence by talking about *strong von-Wright-consistency* [6]. According to Hansen, evaluating permissions only individually is an erroneous approach. Instead, he proposes that regulative norms should be evaluated against all maximal subsets of permissive norms that are jointly consistent with the context.

IO$\top$ coherent *iff* $\forall p \in P(G_R, G_P, A)$: $cons(\{p\} \cup A \cup O(G_R, A))$

IO$\exists$ coherent *iff* $\exists B: \ \forall p \in P(G_R, G_P, A \cup B)$: $cons(\{p\} \cup A \cup B \cup O(G_R, A \cup B))$

IO$\forall$ coherent *iff* $\forall B : cons(A \cup B) \Rightarrow \ (\forall p \in P(G_R, G_P, A \cup B)$: $cons(\{p\} \cup A \cup B \cup O(G_R, A \cup B)))$

IOR coherent *iff* $\forall (a, x) \in G_R \cup G_P: \ \forall p \in P(G_R, G_P, A \cup \{a\})$: $cons(\{p\} \cup A \cup \{a\} \cup O(G_R, A \cup \{a\}))$

We use the notation $coh(G_R, G_P, A)$ *to denote the coherence of* G_R *and* G_P *in context* A. *We write* $\neg coh(G_R, G_P, A)$ *to refer to incoherence, and we may abbreviate* $coh(G_R, \emptyset, A)$ *as* $coh(G_R, A)$, $coh(G_R, \emptyset, \emptyset)$ *as* $coh(G_R)$, *and so on.*

The following example illustrates coherence with both permissive and regulative norms.

Example 6 (Potential conflict between obligations and permissions)**.** Let $G_R = \{(a, x)\}$, $G_P = \{(b, \neg x)\}$ and let O and P be such that $O(G_R, \{x\}) = O(G_R, \{x, b\}) = \emptyset$, $O(G_R, \{x, a\}) = O(G_R, \{x, a, b\}) = \{x\}$, $P(G_R, G_P, A) = \{\neg x\}$ if $b \in A$ (and $\emptyset$ otherwise), $cons(\{a, b, x\})$ and its subsets, and $\neg cons(\{x, \neg x\})$ and its supersets. Then it is the case that $coh^{\top}_{O}(G_R, G_P, \{x\})$, $coh^{\exists}_{O}(G_R, G_P, \{x\})$ and $coh^{R}_{O}(G_R, G_P, \{x\})$, but $\neg coh^{\forall}_{O}(G_R, G_P, \{x\})$, since $cons(\{a, b.x\})$ but $\neg cons(\{\neg x\} \cup O(G_R, \{a, b, x\}))$. Furthermore, we have $coh^{\top}_{IO}(G_R, G_P, \{x\})$ and $coh^{\exists}_{IO}(G_R, G_P, \{x\})$, but $\neg coh^{R}_{IO}(G_R, G_P, \{x\})$, since $\neg cons(\{\neg x, b\} \cup O(G_R, \{x, b\}))$. For the same reason, $\neg coh^{\forall}_{IO}(G_R, G_P, \{x\})$.

Once a normative system is incoherent, you cannot make it coherent by learning new facts:

Proposition 6. *Let* G_R *be a set of regulatory norms, let* G_P *be a set of permissive ones and let* A *be a context:*

$$coh^{Y}_{X}(G_R, G_P, A \cup B) \rightarrow coh^{Y}_{X}(G_R, G_P, A) \text{ with respect to } O^{x}_{i} \text{ and } SP^{x}_{i}$$

Equivalently:

$$\neg coh^{Y}_{X}(G_R, G_P, A) \rightarrow \neg coh^{Y}_{X}(G_R, G_P, A \cup B)$$

Proof. This follows from the fact that O^{x}_{i} and SP^{x}_{i} are monotonic (in all arguments, in particular the last argument representing the context). A similar line of reasoning to that presented in Proposition 5 can be applied. □

6 Properties

We are not only interested in the existence of different deontic operators and the way they are defined, we are also interested in their interaction. This section will analyse how obligation, weak permission and strong permission are related to one another.

Let us formally define the notion of cross-coherence used in [16]. The definition is an adaptation to fit a wider variety of input/output logics. Although we consider many different notions of coherence, as presented in section 2, we also want to compare these with the work previously carried out by Makinson and Van der Torre. Intuitively, a set of regulative norms G_R is cross-coherent with a set of permissive norms if no obligation outputted in a given (consistent) context "negates" a (strong) permission outputted in the same context.

Definition 8 (Cross-coherence)**.** *A set of regulative norms G_R is cross-coherent with a set of permissive norms G_P with respect to I/O operation O_i^ν iff there is no c, u, v with c being classically consistent, $u \wedge v$ being inconsistent, $(c, u) \in O_i^\nu(G_R)$ and $(c, v) \in SP_i^\nu(G_P, G_R)$*

Remark 1. Cross-coherence corresponds to the following notion in our terminology: G_R is cross-coherent with G_P iff $\forall A$ with $A \not\vdash \bot$, it holds that $coh_O^\top(G_R, G_P, A)$, i.e. $\forall A$ with $A \not\vdash \bot$, it holds that $\forall p \in SP_i^\nu(G_R, G_P, A) : cons(\{p\} \cup O_i^\nu(G_R, A))$.

For the unconstrained I/O logics $O_5^* - O_8^*$, Makinson and Van der Torre [16] proved the following:

$$O_i^*(G_R) \subseteq SP_i^*(G_P, G_R) \subseteq^c WP_i^*(G_R) \text{ iff } G_R \text{ and } G_P \text{ are cross-coherent}$$

where $\subseteq^c$ is an *almost-inclusion* ($A \subseteq^c B$ iff whenever $(a, x) \in A$ with a being consistent, then $(a, x) \in B$).

Notice that here, a certain notion of coherence is taken as an assumption. The cross-coherence of G_R and G_P implies this nice set of inclusions, which tells us much about the behaviour of the different operators.

How do these inclusions fare for the other input/output logics? Below are generalised results that hold for all the input/output logics considered.

Proposition 7. *Assume $coh_X^Y(G_R, G_P, A)$ for O_i^ν and SP_i^ν where, as before, $X \in \{I, IO\}$, $Y \in \{\top, \exists, \forall, R\}$ and $i \in \{1, ..., 4\}$. Then:*

$$O_i^\nu(G_R, A) \subseteq SP_i^\nu(G_R, G_P, A) \subseteq WP_i^\nu(G_R, A)$$

Proof. We have that $O_i^\nu(G_R, A) \subseteq SP_i^\nu(G_P, G_R, A)$ since for all the output operations we consider, O_i^ν, are monotone in the first argument, and strong permission is defined as $SP_i^\nu(G_P, G_R, A) = O_i^\nu(G_R \cup Q, A)$ for some $Q \subseteq G_P$, whether singleton or empty. So $O_i^\nu(G_R) \subseteq SP_i^\nu(G_P, G_R)$ follows from the definitions (independently of any coherence-condition).

For the second inclusion, assume $coh_X^Y(G_R, G_P, A)$ and $x \in SP_i^\nu(G_R, G_P, A)$. Reasoning by contradiction, assume $x \notin WP_i^\nu(G_R, A)$, i.e. $\neg x \in O_i^\nu(G_R, A)$. Since the O_i^ν we consider are all monotone, we then get that $\neg cons(\{p\} \cup C \cup O_i^\nu(G_R, A \cup D))$, where $C \in \{\emptyset, A, A \cup B, a : A \cup B \not\vdash \bot, (a, x) \in G_R\}$ and $D \in \{\emptyset, B, a : A \cup B \not\vdash \bot, (a, x) \in G_R\}$. This covers all the definitions of coherence from Definition 7. Hence, $\neg coh_X^Y(G_R, G_P, A)$, which contradicts the initial assumption, so $x \in WP_i^\nu(G_R, A)$. □

From Remark 1 and Proposition 7, the following corollary follows.

Corollary 2. *If G_R and G_P are cross-coherent, then*

$$O_i^\nu(G) \subseteq SP_i^\nu(G, P) \subseteq WP_i^\nu(G)$$

Remark 2. The results for Makinson and Van der Torre, as mentioned above in Proposition 7, hold for the original input/output logics and an "iff" condition of cross-coherence. For O_1–O_3, the "only if" direction does not hold. To see this, consider the following counterexample: $G_R = \{(a, x)\}$, $G_P = \{(a, \neg x \wedge y)\}$. Then G_R is not cross-coherent with G_P, but $SP_i(G_P, G_R) \subseteq^c WP_i(G_R)$. This is because of the lack of WO for permission and the presence of the consistency constraint.

Proposition 8. *Suppose that O_i^ν and O_j^y are such that for any G_R, we have $O_i^\nu(G_R) \subseteq O_j^y(G_R)$. Then the following holds:*

1. $SP_i^\nu(G_R, G_P) \subseteq SP_j^y(G_R, G_P)$
2. $WP_j^y(G_R) \subseteq WP_i^\nu(G_R)$

Proof. 1. Immediate from the definition of SP.

2. Take $(a, x) \in WP_j^y(G_R)$. This means that $(a, \neg x) \notin O_j^y(G_R)$. Since $O_i^\nu(G_R) \subseteq O_j^y(G_R)$, it is also the case that $(a, \neg x) \notin O_i^\nu(G_R)$. Hence $(a, x) \in WP_i^\nu(G_R)$.

□

The results from this section are summarised in Table 4.

Table 4: The relations between the obligation and permission operations studied where $x \in \{\llcorner\!\lrcorner, *\}$, $i = 1, ...8$, G_R is a set of regulatory norms and G_P is a set of permissive norms

$O_i^\nu(G_R) \subseteq SP_i^\nu(G_R, G_P) \subseteq WP_i^\nu(G_R)$	if $coh_X^Y(G_R, G_P, A)$
$SP_i^\nu(G_R, G_P) \subseteq SP_j^y(G_R, G_P)$	if $O_i^\nu \subseteq O_j^y$
$WP_j^y(G_R) \subseteq WP_i^\nu(G_R)$	if $O_i^\nu \subseteq O_j^y$

7 Prospects

Two issues of a more programmatic nature are discussed: obligation under exception, and deontic explanations. It is not our intention to give a complete analysis of these issues, but rather to identify promising directions for future work.

7.1 Obligation with exceptions

At this point, we would like to return to the notion of obligation: can we define an obligation in terms of the triple (G_R, G_P, A) instead of only the pair (G_R, A), as we have done until now? That is, in addition to the generator set G_R and the context A, can we add a set of permissive norms G_P into the notion of obligation? Since explicit permissions often represent exceptions to more general prohibitions, we can do this through the idea of permission as exception, and define an *obligation with exceptions*.

Definition 9 (Obligation with exceptions)**.** *Let G_R be a set of regulative norms, let G_P be a set of permissive norms, A a context, O_i^ν an obligation operation and SP_i^ν the corresponding strong permission operation. Then something is obligatory with exceptions iff it is obligatory in the traditional sense, and if its opposite is not strongly permitted:*

$$x \in O_i^\nu(G_R, G_P, A) \textit{ iff i) } x \in O_i^\nu(G_R, A) \textit{ and ii) } \neg x \notin SP_i^\nu(G_R, G_P, A)$$

Example 7 (Parking regulations)**.** Let $G_R = \{(\top, \neg p)\}$, $G_P = \{(w, p)\}$, where p stands for parking, w stands for weekend and m stands for Monday. G_R and G_P express that it is prohibited to park except on the weekend. Assume that $\neg p \in O_i^\nu(G_R, \top)$, $\neg p \in O_i^\nu(G_R, m)$, $p \in SP_i^\nu(G_R, G_P, w)$ and $p \notin SP_i^\nu(G_R, G_P, m)$. Then we have the following:

- It is not obligatory-with-exceptions to refrain from parking on the weekend: $\neg p \notin O_i^\nu(G_R, G_P, w)$, since $p \in SP_i^\nu(G_R, G_P, w)$
- It is obligatory-with-exceptions to refrain from parking on a Monday: $\neg p \in O_i^\nu(G_R, G_P, m)$, since $p \notin SP_i^\nu(G_R, G_P, m)$

As the example illustrates, looking at obligation in this manner adds a layer of non-monotonicity to the *a priori* monotonic input/output logics: in general, parking is prohibited, but on the weekend the prohibition can no longer be derived.

Remark 3. One could consider using weak permission instead of strong permission in the definition of obligation-with-exceptions. However, doing so has the drawback that obligation with exception is the same as a regular obligation: saying that $\neg x \notin WP_i^\nu(G_R, A)$ is exactly the same as saying that $x \in O_i^\nu(G_R, A)$ by definition of weak permission.

For obligations with exceptions, we have that permissive norms can make an incoherent system coherent by overriding regulative norms, but they cannot make a coherent system incoherent. Formally:

Proposition 9. *Let G_R be a set of regulatory norms, G_P a set of permissive norms and A a context. Then, once a normative system is coherent with respect to obligations with exceptions and strong permissions, we cannot make it incoherent by adding permissive norms.*

$coh_X^Y(G_R, G_P, A) \rightarrow coh_X^Y(G_R, G_P \cup Q, A)$ with respect to $O_i^\nu(G_R, G_P, A)$ and $SP_i^\nu(G_R, G_P, A)$

Proof. Since strong permission $SP_i^\nu(G_R, G_P, A)$ is monotonic, we have that $SP_i^\nu(G_R, G_P, A) \subseteq SP_i^\nu(G_R, G_P \cup Q, A)$. Hence, we have the following inclusion: $O_i^\nu(G_R, G_P, A) \supseteq O_i^\nu(G_R, G_P \cup Q, A)$, since $O_i^\nu(G_R, G_P, A) = \{x : x \in O_i^\nu(G_R, A) \textit{ and } \neg x \notin SP_i^\nu(G_R, G_P, A)\}$ and $O_i^\nu(G_R, G_P \cup Q, A) = \{x : x \in O_i^\nu(G_R, A) \textit{ and } \neg x \notin SP_i^\nu(G_R, G_P \cup Q, A)\}$.

Assume $coh_O^\top(G_R, G_P, A)$. This means that $\forall p \in SP_i^\nu(G_R, G_P, A)$, we have $cons(\{p\} \cup O_i^\nu(G_R, G_P, A))$. Since we have $O_i^\nu(G_R, G_P, A) \supseteq O_i^\nu(G_R, G_P \cup Q, A)$, we get that if $cons(O_i^\nu(G_R, G_P, A))$, then also $cons(O_i^\nu(G_R, G_P \cup Q, A))$. This can be extended to saying that if $cons(\{p\} \cup O_i^\nu(G_R, G_P, A))$, then also $cons(\{p\} \cup O_i^\nu(G_R, G_P \cup Q, A))$. The same can be checked for other notions of coherence.

An axiomatic investigation of the obligation-with-exceptions falls outside the scope of this work, and must be left as a topic for future research.

7.2 Explanation

7.2.1 Coherence explanation

Suppose we need to explain why a normative system with permissive and regulative norms is or isn't coherent in a particular context. The theory developed in the previous subsection offers us a lot of flexibility, depending on the application and audience of the explanation. More precisely, there are five choices to be made in providing the explanation. First, a logic O, a logic P, and a notion of consistency *cons* must be chosen. Then, we need to choose one of the eight notions of coherence by selecting either the output or the input/output constraint, and by selecting one of the four quantifications over contexts.

Example 8 (Deontic explanation)**.** Consider normative system $G_R = \{(a, x), (b, x)\}$, $G_P = \{(c, y), (y, \neg x)\}$ with $A = \{a \vee b, c, \neg x\}$. To explain why this normative system is incoherent, we can choose a notion of consistency where $\{x, \neg x\}$ is inconsistent, a logic O that detaches x, and either one of the following explanations:

Explanation 1 A logic P that detaches $\neg x$, so that we can combine permissive norms with any kind of consistency constraint and quantification over contexts. The reasons are $(a, x), (b, x) \in G_R$, $(c, y), (y, \neg x) \in G_P$ and $a \vee b, c \in A$.

Explanation 2 The input/output constraint, with any kind of logic P and quantification over contexts. The reasons are $(a, x), (b, x) \in G_R$ and $a \vee b, \neg x \in A$.

Which deontic explanation is better depends on the application and audience.

7.2.2 Role of permission in deontic explanations

To explain why something is permitted, we can choose an input/output logic, just like we can to explain an obligation. In addition, we can choose the notion of permission. Moreover, if we choose strong permission, we can also choose a derivation.[16] As studied in the previous subsection, and as summarised in Table 4, if the normative system is cross-coherent, then we have the following:

- The strongest explanation for showing that something is permitted is to show that it is obligatory. The intermediate explanation for showing that something is permitted is to show that it is strongly permitted. And the weakest kind of explanation is to show that it is weakly permitted.
- The effect of the choice of input/output logic on the permissions is given in the other two rows: if a stronger input/output logic is chosen, then we can derive more strong permissions but fewer weak permissions.

Though the study of deontic explanations is still at an initial phase, we can already see that permissive norms can play a central role in deontic explanations. If we want to show that something is not obligatory, we can either use the semantics to compute everything that is obligatory (as discussed in Section 3.3) or we can use weak permissions—but then we cannot use a derivation. So the best way is to use permissive norms.[17]

A thorough investigation of the role of permission in deontic explanations is yet to be carried out.

8 Some related work

In our work, the notion of coherence has been used as an assumption, but other approaches illustrate that it can be used in different ways. For instance, coherence can also be used as a *constraint*. A notable example of such usage is provided in constrained input/output logic [15], where the procedure for generating outputs takes an additional consistency parameter with which the output has to be consistent (the output consistency and input/output consistency that we studied are two special cases of such a consistency constraint.) The procedure continues by taking all the maximal subsets of the normative system that are consistent with the consistency parameter, and finishes by taking the meet or join of the outputs of these subsets. These outputs of the subsets are called outfamilies, corresponding to extensions in formalisms like logic programming and default logic [31].[18] A question related to this approach is how to define

[16] Though we do not discuss dynamic permissions in this article, we observe that they may also play an important role in deontic explanations because for traditional input/output logics, they satisfy the same rules as weak permission.

[17] Moreover, given the central role of permissive norms in the deontic explanations of permissions, and for concrete applications in which no permissive norms are given, one might try to guess permissive norms for deontic explanations.

[18] These outfamilies or extensions can be used for an alternative notion of coherence: (G_R, G_P, A) is coherent if and only if the number of extensions of (G_R, A) is the same as the number of extensions of $(G_R \cup G_P, A)$. The extension approach is especially interesting because it does not limit itself to giving a binary coherent/incoherent answer. Instead, by providing a differing number of extensions, it allows us to reason with *degrees of coherence*; the bigger the differing number of extensions, the more the system is incoherent.

permissions under constraints. It seems that the set of permissions will change dramatically depending on whether one takes the meet or the join of the family of outputs. This needs to be analysed.

Makinson and Van der Torre studied not only weak and strong permission, but also a notion in between which they called *positive dynamic permission*. We have not studied dynamic permission in this article, but such an analysis would be interesting, especially since dynamic permission seems to combine both underivability and derivability, which are distinguishing notions of weak and strong permission. Furthermore, [16] provides characterisation results for some of the original input/output logics for dynamic permission, but it is not clear how these results can extend to other logics.

Hansen argues that permissions should also be allowed some restricted form of aggregations [6]. He argues that permissions should be taken into account collectively as long as they remain consistent. For instance, in the drinking-and-driving example, they should not be allowed to aggregate, however permissions for drinking and taking a taxi should be allowed to aggregate. This is an interesting notion which merits further investigation.

Tossato et al. [33, 34] represent normative systems in a graph-based manner. They provide a graphical framework to discuss obligations, permissions and constitutive norms on an abstract level. Their framework could potentially be used for deontic explanation and should be further investigated in that context.

9 Conclusion

In this work, we studied the notion of permission for the whole family of available input/output logics. We provided a comprehensive overview of sixteen axiomatic systems for input/output logics. As opposed to the standard way of representing input/output logic semantically, we chose to focus on an axiomatic representation as it is more reader-friendly and intuitive. We also introduced a new input/output logic O_2 with a built-in consistency constraint which supports a restricted form of aggregation and a generalised form of reasoning by cases. We also provided soundness and completeness results. On this basis, we carried out a systematic analysis of weak and strong permissions for the logics presented before. We provided sets of rules for negative and positive permission, soundness and completeness results for strong permission for seven of the studied input/output logics and discussed the remaining logics that are not complete. We also argued that allowing weakening of the output might not be desirable when dealing with permissions with inseparable content, and that consistency constraints are desirable when dealing with contrary-to-duty reasoning. Furthermore, we studied the relations between the operators studied in this article: obligation, weak permission and strong permission. Lastly, we introduced obligation with exceptions, which adds a layer of non-monotonicity to the otherwise monotonic input/output logics. We did this by stating that something is obligatory with exceptions if that follows from the monotonic obligation operation and if its negation is not (strongly) permitted.

References

[1] D. A. Ambrossio, X. Parent, and L. van der Torre. Cumulative aggregation. In O. Roy, A. Tamminga, and M. Willer, editors, *Deontic Logic and Normative Systems*, pages 1–15, London, 2016. College Publications.

[2] G. Boella and L. van der Torre. Institutions with a hierarchy of authorities in distributed dynamic environments. *Artificial Intelligence and Law*, 16(1):53–71, Mar 2008.

[3] A. Chopra, L. van der Torre, and H. Verhagen. *Handbook of Normative Multiagent Systems*. College Publications, 2018.

[4] A. Farjami. *Discursive Input/Output Logic: Deontic Modals, and Computation*. PhD thesis, University of Luxembourg, Luxembourg, 2020.

[5] F. Feldman. *Doing the Best We Can: An Essay in Informal Deontic Logic*. D. Reidel Publishing Company, Dordrecht, 1986.

[6] J. Hansen. Reasoning about permission and obligation. In S. O. Hansson, editor, *David Makinson on Classical Methods for Non-Classical Problems*, pages 287–333. Springer Netherlands, Dordrecht, 2014.

[7] J. Hansen, G. Pigozzi, and L. van der Torre. Ten philosophical problems in deontic logic. In G. Boella, L. van der Torre, and H. Verhagen, editors, *Normative Multi-agent Systems*, number 07122 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[8] S. O. Hansson. The varieties of permission. In D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*, pages 195–240. College Publications, London, 2013. Volume 1.

[9] R. Hilpinen and P. McNamara. Deontic logic: a historical survey and introduction. In D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*, volume 1, pages 3–136. College Publications, London, 2013.

[10] J. Jørgensen. Imperatives and logic. *Erkenntnis*, 7(1):288–296, 1937.

[11] B. Lellmann. From input/output logics to conditional logics via sequents - with provers. In A. Das and S. Negri, editors, *Automated Reasoning with Analytic Tableaux and Related Methods - 30th International Conference, TABLEAUX 2021*, volume 12842 of *Lecture Notes in Computer Science*, pages 147–164. Springer, 2021.

[12] B. Lellmann, F. Gulisano, and A. Ciabattoni. Mīmāṃsā deontic reasoning using specificity: a proof theoretic approach. *Artificial Intelligence and Law*, 29:351– 394, 9 2021.

[13] B. Lellmann and D. Pattinson. Sequent systems for Lewis' conditional logics. In L. F. del Cerro, A. Herzig, and J. Mengin, editors, *Logics in Artificial Intelligence*, pages 320–332, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[14] D. Makinson and L. van der Torre. Input/output logics. *Journal of Philosophical Logic*, 29(4):383–408, 2000.

[15] D. Makinson and L. van der Torre. Constraints for input/output logics. *Journal of Philosophical Logic*, 30(2):155–185, 2001.

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

[17] R. B. Marcus. Moral dilemmas and consistency. *The Journal of Philosophy*, 77(3):121–136, 1980.

[18] M. Olszewski, X. Parent, and L. van der Torre. Input/output logic with a consistency check - the case of permission. In F. Liu, A. Marra, P. Portner, and F. V. D. Putte, editors, *Deontic Logic and Normative Systems. DEON 2020/2021.* College Publications, 2021.

[19] X. Parent. A modal characterisation of an intuitionistic I/O operation. *FLAP*, 8(8):2349–2362, 2021.

[20] X. Parent, D. Gabbay, and L. van der Torre. Intuitionistic basis for input/output logic. In S. O. Hansson, editor, *David Makinson on Classical Methods for Non-Classical Problems*, pages 263–286. Springer Netherlands, Dordrecht, 2014.

[21] X. Parent and L. van der Torre. Input/output logic. In D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*, volume 1, chapter 8, pages 499–544. College Publications, London, 2013.

[22] X. Parent and L. van der Torre. "Sing and Dance!". In F. Cariani, D. Grossi, J. Meheus, and X. Parent, editors, *Deontic Logic and Normative Systems, DEON 2014*, pages 149–165, Cham, 2014. Springer.

[23] X. Parent and L. van der Torre. The pragmatic oddity in norm-based deontic logics. In G. Governatori, editor, *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, page 169–178, New York, NY, USA, 2017. Association for Computing Machinery.

[24] X. Parent and L. van der Torre. Detachment in normative systems: Examples, inference patterns, properties. In A. Chopra, L. van der Torre, and H. Verhagen, editors, *Handbook of Normative Multiagent Systems*, pages 305–343. College Publications, 2018.

[25] X. Parent and L. van der Torre. *Introduction to Deontic Logic and Normative Systems.* Texts in logic and reasoning. College Publications, 2018.

[26] X. Parent and L. van der Torre. I/O logics with a consistency check. In J. Broersen, C. Condoravdi, S. Nair, and G. Pigozzi, editors, *Deontic Logic and Normative Systems*, pages 285–299, London, 2018. College Publications.

[27] X. Parent and L. van der Torre. Input/output logics without weakening. *Filosofiska Notiser*, 6(1):189–208, 2019.

[28] X. Parent and L. van der Torre. I/O logics without weakening. *Filosofiska Notiser*, 6:189–208, 2019.

[29] D. Pedreschi, F. Giannotti, R. Guidotti, A. Monreale, S. Ruggieri, and F. Turini. Meaningful explanations of black box AI decision systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:9780–9784, 07 2019.

[30] H. Prakken and M. Sergot. Dyadic deontic logic and contrary-to-duty obligations. In D. Nute, editor, *Defeasible Deontic Logic*, pages 223–262. Kluwer Academic Publishers, Dordrecht, 1997.

[31] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1):81–132, 1980. Special Issue on Non-Monotonic Logic.

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

[33] S. C. Tosatto, G. Boella, L. van der Torre, and S. Villata. Abstract normative systems: Semantics and proof theory. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 358–368. AAAI Press, 2012.

[34] S. C. Tosatto, G. Boella, L. van der Torre, and S. Villata. Visualizing normative systems: An abstract approach. In T. Ågotnes, J. M. Broersen, and D. Elgesem, editors, *Deontic Logic in Computer Science - 11th International Conference, DEON 2012, Bergen, Norway, July 16-18, 2012. Proceedings*, volume 7393 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 2012.

[35] K. van Berkel, A. Ciabattoni, E. Freschi, F. Gulisano, and M. Olszewski. The gentle murder paradox in sanskrit philosophy. In F. Liu, A. Marra, P. Portner, and F. V. D. Putte, editors, *Deontic Logic and Normative Systems, DEON2021.* College Publications, 2021.

[36] G. von Wright. Deontic Logic. *Mind*, 60:1–15, 1951. reprinted in [38], pages 58–74.

[37] G. von Wright. *Norm and Action: A Logical Enquiry*. International Library of Philosophy and Scientific Method. Routledge & Kegan Paul, 1963.

[38] G. H. von Wright, editor. *Logical Studies*. Routledge and Kegan Paul, London, 1957.

A Semantics

All the logics defined in this appendix are sound and complete with respect to the corresponding rules from Table 1. For O_2, the semantic definition is novel, and soundness and completeness are provided in Section 3.

A.1 Operations O_1^*-O_3^* [22]

Definition 10. *We define:*

- $x \in O_1^*(G_R, A)$ *iff there is a finite* $M \subseteq G_R$ *such that i)* $M(Cn(A)) \neq \emptyset$ *and ii)* $x \dashv\vdash \bigwedge M(Cn(A))$
- $O_2^*(G_R, A) = \cap\{O_1^*(G_R, V) : A \subseteq V \ , V \text{ complete}\}$
  where $V \subseteq \mathcal{L}$ *is complete iff* $V = \mathcal{L}$ *or* V *is maximal consistent.*
- $x \in O_3^*(G_R, A)$ *iff there is a finite* $M \subseteq G_R$ *such that i)* $M(Cn(A)) \neq \emptyset$ *and ii)* $\forall B$ *such that* $A \subseteq B = Cn(B) \supseteq M(B)$, $x \dashv\vdash \bigwedge M(B)$

A.2 Operations O_5^*–O_8^* [14]

Definition 11. *We define:*

- $O_5^*(G_R, A) = Cn(G_R(Cn(A)))$
- $O_6^*(G_R, A) = \cap(Cn(G_R(V)) : A \subseteq V, V complete)$
- $O_7^*(G_R, A) = \cap(Cn(G_R(B)) : A \subseteq B = Cn(B) \supseteq G_R(B))$
- $O_8^*(G_R, A) = \cap(Cn(G_R(V)) : A \subseteq V \supseteq G_R(V), V complete)$

where $V \subseteq \mathcal{L}$ *is complete iff* $V = \mathcal{L}$ *or* V *is maximal consistent.*

A.3 Operations O_1–O_3 [24]

Definition 12. *We define:*

- $x \in O_1(G_R, A)$ *iff there exists some finite* $M \subseteq G_R$ *and a set* $B \subseteq Cn(A)$ *such that* $M \neq \emptyset$, $B = b(M)$, $x \dashv\vdash \wedge h(M)$ *and* $\{x\} \cup B$ *is consistent.* $O_1(G_R) = \{(A, x) : x \in O_1(G_R, A)\}$.
- $x \in O_3(G_R, A)$ *iff there exists some finite* $M \subseteq G_R$ *and a set* $B \subseteq Cn(A)$ *such that* $M(B) \neq \emptyset$, $x \dashv\vdash \wedge h(M)$ *and*
  - $\forall B'(B \subseteq B' = Cn(B') \supseteq M(B') \Rightarrow b(M) \subseteq B')$
  - $\{x\} \cup B$ *is consistent.*

B Proof of NRP

We now turn to the proof of Proposition 4. For D_5^*, D_6^* and D_7^*, proofs were provided by Makinson and Van der Torre [16] (note that in that work, the I/O operations were called out_1, out_2 and out_3 respectively).

Proposition 10. *D_1^* satisfies NRP for rules EQ, SI, AND.*

Proof. We prove this by showing that for every $(b, y) \in D_1^*(G_R)$, we can construct a derivation in $D_1^*(G_R)$ in a certain way such that every leaf is only used once. The construction is essentially the same as in the completeness proof of D_1^* in [22] (note that in that work, O_1^* was called O_1).

Take $(b, y) \in D_1^*(G_R)$. By soundness, $(b, y) \in O_1^*(G_R)$, and by the definition of O_1^*, this means that there exists a finite witness $W \subseteq G_R$ such that $W(Cn(b))$ is non-empty and $y \dashv\vdash \bigwedge W(Cn(b))$. That is, $y \dashv\vdash x_1 \wedge ... \wedge x_n$ such that $(a_i, x_i) \in W$ with $a_i \in Cn(b)$.

Let $L = \{(a_1, x_1), ..., (a_n, x_n)\}$ be the enumeration of these norms. These shall be the leaves of our derivation. Then, we can construct the derivation of (b, y), using each element of L only once, in the following way:

$$\dfrac{\dfrac{\dfrac{(a_1, x_1)}{(b, x_1)}\text{SI} \quad ... \quad \dfrac{(a_n, x_n)}{(b, x_n)}\text{SI}}{(b, \bigwedge_{i=1}^{n} x_i)}\text{AND}}{(b, y)}\text{EQ}$$

□

Proposition 11. *D_3^* satisfies NRP for rules EQ, SI, ACT.*

Proof. The proof here is omitted as it is similar to the proof of D_7^* from [16] (it is called $deriv_3$ in that work). In [16] in the Lemmas 3.3.1-3.3.3, replacing WO with EQ and removing TAUT (which, is not a problem, as TAUT does not play any particular role in these results) will give us the proof we need. □

Proposition 12. *D_1 satisfies NRP for rules EQ, SI, R-AND.*

Proof. This proof is very similar to the proof of NRP for D_1^* in Proposition 10. Here again, we prove the proposition by constructing a derivation.

Take $(b, y) \in D_1(G_R)$. By soundness, $(b, y) \in O_1(G_R)$. By the definition of O_1, there exists a non-empty witness $W \subseteq G_R$ and $B \subseteq Cn(b)$ with $B = bodies(W)$, $y \dashv\vdash \bigwedge heads(W)$ and $\{y\} \cup B \nvdash \bot$.

Let $W = \{(a_1, x_1), ..., (a_n, x_n)\}$ be an enumeration of the elements in the witness. Then $\bigwedge_{i=1}^{n} a_i \wedge x_i \dashv\vdash \bigwedge B \wedge y \nvdash \bot$, and thus the consistency check for the application of the restricted aggregation rule R-AND is satisfied.

We can then construct the following derivation:

$$\dfrac{\dfrac{(a_1, x_1)}{(b, x_1)}\text{SI} \qquad \dots \qquad \dfrac{(a_n, x_n)}{(b, x_n)}\text{SI}}{\dfrac{(b, \bigwedge_{i=1}^{n} x_i)}{(b, y)}\text{EQ}}\text{R-AND}$$

□

Proposition 13. *D_3 satisfies NRP for rules EQ, SI, R-ACT.*

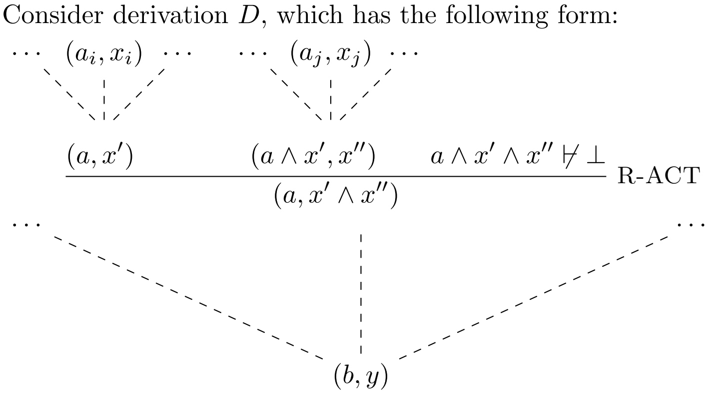
Lemma 2. *Let D be a derivation in D_3, i.e. using the rules EQ, SI, R-ACT. Then at any line (a, x) of the derivation:*

- *$x \vdash x'$ for any (a', x') that is used in D to obtain (a, x);*
- *$a \wedge x \vdash a'$ and $a \wedge x \vdash x'$ for any (a', x') that is used in D to obtain (a, x).*

The proof is a straightforward proof by induction on the length of the derivation, and is omitted here.

Lemma 3. *Let D be a derivation of (b, y) from leaf set L in D_3 (i.e. using the rules EQ, SI, R-ACT), and let (b', y') be the conclusion of an R-ACT application in D. Then there exists a derivation D' of (b, y) from leaf set L, which is like D except that the two sub-derivations that lead to (b', y') follow the order SI, R-ACT, EQ.*

Consider derivation D, which has the following form:



In this instance, (b', y') is $(a, x' \wedge x'')$. Let d_1 be the left sub-derivation, and d_2 the right sub-derivation, with (a, x') and $(a \wedge x', x'')$ as their respective roots and $L(d_1)$, $L(d_2)$ as leaf sets. We will show that there are derivations d_1' and d_2' that lead to the same conclusions as d_1 and d_2 ((a, x') and $(a \wedge x', x'')$ respectively), but follow the order SI, R-ACT, EQ. The EQ rule is invertible both with SI and R-ACT, and can be applied at any point in the derivation. Without loss of generality, assume that EQ is applied at the bottom of derivations d_1 and d_2. That leaves rules SI and R-ACT above in the upper parts of d_1 and d_2. By Lemma 2, it holds that $a \wedge x' \wedge x'' \vdash a_k$ and $a \wedge x' \wedge x'' \vdash x_k$ for every norm in d_1 and d_2. This provides that in d_1 and d_2, R-ACT followed by SI can be inverted to SI followed by R-ACT. As such, the following derivation:

$$\dfrac{\dfrac{(c, y_1) \qquad (c \wedge y_1, y_2) \qquad c \wedge y_1 \wedge y_2 \nvdash \bot}{(c, y_1 \wedge y_2)}\text{R-ACT}}{(d, y_1 \wedge y_2)}\text{SI}$$

can be transformed into:

$$\dfrac{\dfrac{(c, y_1)}{(d, y_1)}\text{SI} \qquad \dfrac{(c \wedge y_1, y_2)}{(d \wedge y_1, y_2)}\text{SI} \qquad d \wedge y_1 \wedge y_2 \nvdash \bot}{(d, y_1 \wedge y_2)}$$

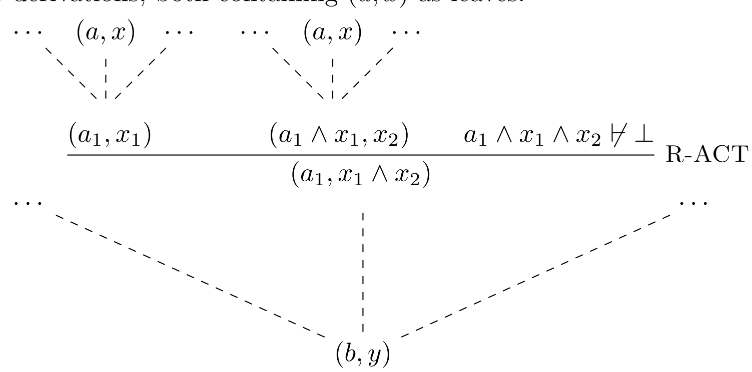
$d \wedge y_1 \wedge y_2 \nvdash \bot$ follows from:

- $a \wedge x' \wedge x'' \vdash d$ (Lemma 2)
- $a \wedge x' \wedge x'' \vdash y_1$ (Lemma 2)
- $a \wedge x' \wedge x'' \vdash y_2$ (Lemma 2)
- $a \wedge x' \wedge x'' \nvdash \bot$ (R-ACT assumption)

which means that the derivations d_1 and d_2 can be phased to SI, R-ACT, EQ. □

We can now prove Proposition 13.

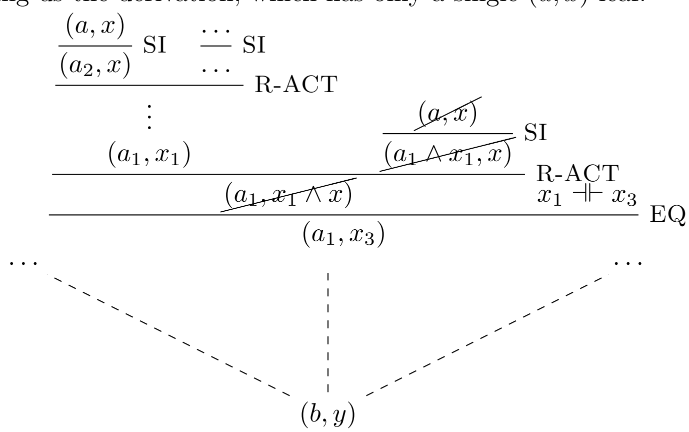
Proof. Looking at derivations having two (a, x) leaves, we are in the following scenario, with $(a_1, x_1 \wedge x_2)$ being the conclusion of the meeting point of two sub-derivations, both containing (a, x) as leaves:



By Lemma 3, we know that those two sub-derivations can be replaced by derivations where the order of the rules is SI, R-ACT, EQ.

The rest of the proof is similar to the proof in Observation 3(c) provided by Makinson and Van der Torre [16]. The R-ACT rule goes from (a, x), $(a \wedge x, y)$ with $a \wedge x \wedge y \nvdash \bot$ to $(a, x \wedge y)$. We call (a, x) the *minor premise* and $(a \wedge x, y)$ the *major premise*. The succession of R-ACT can be written so that no major premise of an application of R-ACT is the conclusion of another application of R-ACT. This has been shown for ACT [16] (from (a, x) and $(a \wedge x, y)$ to $(a, x \wedge y)$), and it still holds for its restricted version.

Since $(a_1 land x_1, x_2)$ is a major premise of R-ACT, it is not the conclusion of another R-ACT application, which means that it follows only from applications of SI, and hence has only leaf (a, x). On the other hand, (a_1, x_1) follows from applications of SI and R-ACT. Lemma 2 give us that $x_2 \dashv\vdash x$ and $x_1 \vdash x$. So $x_1 \dashv\vdash x_1 \wedge x$, and one can remove the subtree with conclusion $(a_1 \wedge x_1, x_2)$, giving us the derivation, which has only a single (a, x) leaf:



□

C Rules

This section provides an overview of the different inference rules used. The superscript o means that the norm is an obligation, and the supersript p means it is a permission.

1. Strengthening of the input

   - Rule

$$\frac{(a, x)^o \qquad b \vdash a}{(b, x)^o}\ \text{SI}$$

   - Subverse rule

$$\frac{(a, x)^p \qquad b \vdash a}{(b, x)^p}\ \text{SI}^{\downarrow}$$

   - Inverse rule

$$\frac{(a, x)^p \qquad a \vdash b}{(b, x)^p}\ \text{SI}^{-1}$$

2. Weakening of the output

   - Rule

$$\frac{(a, x)^o \qquad x \vdash y}{(a, y)^o}\ \text{WO}$$

- Subverse rule

$$\frac{(a,x)^p \qquad x \vdash y}{(a,y)^p}\ \mathrm{WO}^{\downarrow}$$

- Inverse rule

$$\frac{(a,x)^p \qquad x \vdash y}{(a,y)^p}\ \mathrm{WO}^{-1}$$

3. Equivalence of the output

- Rule

$$\frac{(a,x)^o \qquad x \dashv\vdash y}{(a,y)^o}\ \mathrm{EQ}$$

- Subverse rule

$$\frac{(a,x)^p \qquad x \dashv\vdash y}{(a,y)^p}\ \mathrm{EQ}^{\downarrow}$$

- Inverse rule

$$\frac{(a,x)^p \qquad x \dashv\vdash y}{(a,y)^p}\ \mathrm{EQ}^{-1}$$

4. Reasoning by cases

- Rule

$$\frac{(a,x)^o \qquad (b,x)^o}{(a \vee b, x)^o}\ \mathrm{OR}$$

- Subverse rule

$$\frac{(a,x)^p \qquad (b,x)^o}{(a \vee b, x)^p}\ \mathrm{OR}^{\downarrow}$$

- Inverse rule

$$\frac{(a \vee b, \neg x)^p \qquad (a,x)^o}{(b, \neg x)^p}\ \mathrm{OR}^{-1}$$

5. Extended reasoning by cases

- Rule

$$\frac{(a,x)^o \qquad (b,y)^o}{(a \vee b, x \vee y)^o}\ \text{ex-OR}$$

- Subverse rule

$$\frac{(a,x)^p \qquad (b,y)^o}{(a \vee b, x \vee y)^p}\ \text{ex-OR}^{\downarrow}$$

- Inverse rule

$$\frac{(a \vee b, \neg(x \vee y))^p \qquad (a, x)^o}{(b, \neg y)^p} \text{ ex-OR}^{-1}$$

6. Aggregation

- Rule

$$\frac{(a, x)^o \qquad (a, y)^o}{(a, x \wedge y)^o} \text{ AND}$$

- Subverse rule

$$\frac{(a, x)^p \qquad (a, y)^o}{(a, x \wedge y)^p} \text{ AND}^{\downarrow}$$

- Inverse rule

$$\frac{(a, \neg(x \wedge y))^p \qquad (a, y)^o}{(a, \neg x)^p} \text{ AND}^{-1}$$

7. Restricted aggregation

- Rule

$$\frac{(a, x)^o \qquad (a, y)^o \qquad a \wedge x \wedge y \not\vdash \bot}{(a, x \wedge y)^o} \text{ R-AND}$$

- Subverse rule

$$\frac{(a, x)^p \qquad (a, y)^o \qquad a \wedge x \wedge y \not\vdash \bot}{(a, x \wedge y)^p} \text{ R-AND}^{\downarrow}$$

- Inverse rule

$$\frac{(a, \neg(x \wedge y))^p \qquad (a, y)^o \qquad a \wedge x \wedge y \not\vdash \bot}{(a, \neg x)^p} \text{ R-AND}^{-1}$$

8. Cumulative transitivity

- Rule

$$\frac{(a, x)^o \qquad (a \wedge x, y)^o}{(a, y)^o} \text{ CT}$$

- Subverse rules

$$\frac{(a, x)^p \qquad (a \wedge x, y)^o}{(a, y)^p} \text{ CT}^{\downarrow}$$

$$\frac{(a, x)^o \qquad (a \wedge x, y)^p}{(a, y)^p} \text{ CT}^{\downarrow}$$

- Inverse rules

$$\frac{(a,\neg y)^p \qquad (a\wedge x,y)^o}{(a,\neg x)^p}\ \mathrm{CT}^{-1}$$

$$\frac{(a,\neg y)^p \qquad (a,x)^o}{(a\wedge x,\neg y)^p}\ \mathrm{CT}^{-1}$$

9. Aggregative cumulative transitivity

- Rule

$$\frac{(a,x)^o \qquad (a\wedge x,y)^o}{(a,x\wedge y)^o}\ \mathrm{ACT}$$

- Subverse rules

$$\frac{(a,x)^p \qquad (a\wedge x,y)^o}{(a,x\wedge y)^p}\ \mathrm{ACT}^{\downarrow}$$

$$\frac{(a,x)^o \qquad (a\wedge x,y)^p}{(a,x\wedge y)^p}\ \mathrm{ACT}^{\downarrow}$$

- Inverse rules

$$\frac{(a,\neg(x\wedge y))^p \qquad (a\wedge x,y)^o}{(a,\neg x)^p}\ \mathrm{ACT}^{-1}$$

$$\frac{(a,\neg(x\wedge y))^p \qquad (a,x)^o}{(a\wedge x,\neg y)^p}\ \mathrm{ACT}^{-1}$$

10. Restricted aggregative cumulative transitivity

- Rule

$$\frac{(a,x)^o \qquad (a\wedge x,y)^o \qquad a\wedge x\wedge y\nvdash\bot}{(a,x\wedge y)^o}\ \text{R-ACT}$$

- Subverse rules

$$\frac{(a,x)^p \qquad (a\wedge x,y)^o \qquad a\wedge x\wedge y\nvdash\bot}{(a,x\wedge y)^p}\ \text{R-ACT}^{\downarrow}$$

$$\frac{(a,x)^o \qquad (a\wedge x,y)^p \qquad a\wedge x\wedge y\nvdash\bot}{(a,x\wedge y)^p}\ \text{R-ACT}^{\downarrow}$$

- Inverse rules

$$\frac{(a,\neg(x\wedge y))^p \qquad (a\wedge x,y)^o \qquad a\wedge x\wedge y\nvdash\bot}{(a,\neg x)^p}\ \text{R-ACT}^{-1}$$

$$\frac{(a,\neg(x\wedge y))^p \qquad (a,x)^o \qquad a\wedge x\wedge y\nvdash\bot}{(a\wedge x,\neg y)^p}\ \text{R-ACT}^{-1}$$

$$a,x,y\nvdash\bot\ \frac{(a,x) \qquad (a,y)}{(a,x\wedge y)}\ \text{R-AND}$$