Cumulative Aggregation

Diego Agustín Ambrossio Xavier Parent Leendert van der Torre

University of Luxembourg
6, rue Richard Coudenhove-Kalergi, Luxembourg
{diego.ambrossio,xavier.parent,leon.vandertorre}@uni.lu

Abstract

From any two conditional obligations "X if A" and "Y if B", cumulative aggregation derives the combined obligation "$X \cup Y$ if $A \cup (B \setminus X)$", whereas simple aggregation derives the obligation "$X \cup Y$ if $A \cup B$". We propose FC systems consisting of cumulative aggregation together with factual detachment, and we give a representation result for FC systems, as well as for FA systems consisting of simple aggregation together with factual detachment. We relate FC and FA systems to each other and to input/output logics recently introduced by Parent and van der Torre.

Keywords: cumulative aggregation, abstract normative systems, input/output logic.

1 Introduction

In this paper, we contrast and study two different principles of aggregation for norms in the context of the framework of Abstract Normative Systems (ANS) due to Tosatto et al. [9].

This one is intended as a general framework to compare logics for normative reasoning. Only fragments of the standard input/output logics [5] are covered by Tosatto et al., and so here we set ourselves the task of applying the framework to the input/output logic recently introduced by Parent and van der Torre [7]. (Cf. also [6].) Its most salient feature is the presence of a non-standard form of cumulative transitivity, called "aggregative" (ACT, for short). Such a rule is used in order to block the counter-examples usually given to the principle known as "deontic detachment": from the obligation of X and the obligation of Y if X, infer the obligation of Y.

Our contribution is first and foremost technical. We acknowledge that the benefits of using the theory of abstract normative systems may not be obvious to the reader. We will not discuss the question of whether it has a reasonable claim to be a general framework subsuming others, nor will we discuss the question of whether aggregative cumulative transitivity is, ultimately, the right form of transitivity.

A central feature of the Tosatto et al. account is that it abstracts away from the language of propositional logic. We recall that as initially conceived

input/output logic is an attempt to generalize the study of conditional obligation from modal logic to the abstract study of conditional codes viewed as relations between Boolean formulas. The underlying language is taken from propositional logic. It contains truth-functional connectives, and is assumed to be closed under application of these connectives. It is natural to ask if one can extend the generality further, by working with an arbitrary language, viewed as a collection of items, and without requiring that the items under consideration be "given" or regimented in some special way. Similar programs have been run for propositional logic and modal logic. Koslow [4]'s structuralist approach to logic is perhaps one of the best-known examples of such a program. Unlike Koslow, we do not even assume that the items under consideration can enter into some special implication relations with each other. There are scholars who (rightly or wrongly) take the well-known Tarskian conditions for the consequence relation to be objectionable on the grounds that, for reasons of vagueness (or more), important consequence relations over natural languages (however formalized) are, for instance, not generally transitive. (See, e.g., [8].) The idea is just to investigate the possibility of a formal theory of normative reasoning that avoids such commitments (be they justified or not). [1]

Tosatto et al.'s account has no apparatus for handling conjunction of outputs, and our main purpose in this paper is to develop it to do so. We follow the ideas of so-called "multiple-conclusion logic", and treat normative consequence as a relation between sets, whose elements are understood conjunctively. No assumption about the inner structure of these elements is made.

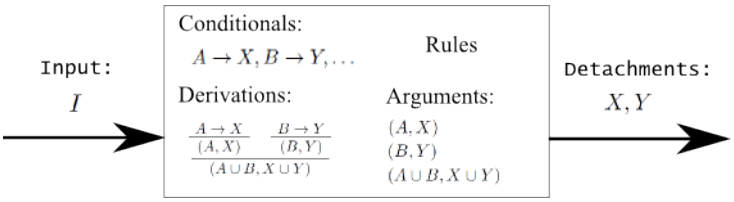

Fig. 1. An Abstract Normative System

An example of an abstract normative system studied in this paper is given in Figure 1. It should be read as follows. Conditionals $A \to X, B \to Y, \ldots$ are the norms of the normative system. Each of A, X, B and Y is a set of language elements (whose inner structure remains unanalyzed). Sets are understood conjunctively on both sides of $\to$. The input I is a collection of language elements representing the context. Rules are used to generate derivations and arguments based on I. The set of detachments $\{X, Y, \ldots\}$ is the output consisting of all detached obligations. The elements of Figure 1 are explained in more detail in the next two sections.

The prime focus in [7] was the contrast between two forms of transitivity, called "cumulative transitivity" and "aggregative cumulative transitivity".

[1] This motivation for using ANS is ours.

This paper shifts the emphasis on the contrast between the following two forms of aggregation.

Simple aggregation If X is obligatory in context A, and Y is obligatory in context B, then $X \cup Y$ is obligatory in context $A \cup B$. In other words, simple aggregation derives the obligation "$X \cup Y$ if $A \cup B$" from any two conditional obligations "X if A" and "Y if B". [2]

Cumulative aggregation If X is obligatory in context A, and Y is obligatory in context B, then $X \cup Y$ is obligatory in context $A \cup (B \setminus X)$. In other words, cumulative aggregation derives the combined obligation "$X \cup Y$ if $A \cup (B \setminus X)$" from the same two conditional obligations.

The rule of simple aggregation gives the most straightforward way of collecting items as detachments are performed. When $A = B$, simple aggregation gives the rule "If X is obligatory given A, and Y is obligatory given A, then $X \cup Y$ is obligatory given A." A drawback of simple aggregation is that it does not capture transitive reasoning. Given the two conditional obligations "$\{x\}$ if $\{\}$" and "$\{y\}$ if $\{x\}$", simple aggregation only yields "$\{x, y\}$ if $\{x\}$". This motivates the rule of cumulative aggregation. In the particular case where $B = A \cup X$, cumulative aggregation yields the form of transitivity introduced by Parent and van der Torre [7] under the name ACT. This is the rule $(A, X), (A \cup X, Y)/(A, X \cup Y)$. In our example, one gets "$\{x, y\}$ if $\{\}$." [3]

To summarize, we adress the following issues:

- How to develop the theory of abstract normative systems to handle conjunction of outputs and the form of cumulative transitivity described in [7]?
- How to define the proof theory of the system? What are the most significant properties of the framework?
- How to provide a semantical characterisation, along with a representation result linking it with the proof theory?

The layout of this paper is as follows. In Section 2, we introduce FA systems for simple aggregation. In Section 3, we introduce FC systems for cumulative aggregation. We give representation results for both systems. In Section 4, we show how FA and FC systems relate to one another, and we discuss some properties of the systems. In Section 5 we show how FA and FC systems relate with the input/output logics introduced by Parent and van der Torre [7].

Due to space limitation, we focus on the logical framework and the results, and leave the proofs of the representation theorems to a technical report [1]. We would like to stress that these are not just a re-run of the proofs given by Parent and van der Torre [7] in a classical logic setting. The two settings

[2] Note that intersection as used in abstract normative systems does not correspond to disjunction in propositional logic. Take $(\{p\}, \{x\})$ and $(\{q\}, \{x\})$. The intersection of the two contexts yields $(\{\}, \{x\})$. Reasoning by cases would yield $(\{p \vee q\}, \{x\})$ instead.

[3] As mentioned, it is not our purpose to discuss this rule in any greater depth. For more details on it, see Parent and van der Torre [7].

are very different. The question of whether the proofs of our representation results can be adapted to yield a completeness result in a classical logic setting remains an open problem.

2 FA systems for simple aggregation

In this section, we introduce abstract normative systems for simple aggregation, and we give a representation result. Though FA systems may be interesting in their own right, in this paper the main role of FA systems is to set the stage for FC systems for cumulative aggregation, introduced in the next section. Thus, although we talk about normative systems and use examples from normative system it must be kept in mind that FA systems are not appropriate for all kinds of normative reasoning.

In general, a system $\langle L, C, R\rangle$ consists of a language L, a set of conditionals C defined over this language, and a set of rules R. The input is a set of sentences from L. If $\langle L, C, R\rangle$ is a normative system, then a conditional $A \to X$ can be read as the norm "if A, then obligatory X". A normative system contains at least one set of norms, the regulative norms from which obligations and prohibitions can be detached. It may also contain permissive norms, from which explicit permissions can be detached, and constitutive norms, from which institutional facts can be detached. In this paper we do not consider permissive and constitutive norms. In the present setting, a system generates or produces an obligation set, a subset of the universe, reflecting the obligatory elements of the universe.

All abstract normative systems we consider satisfy at least factual detachment. To represent factual detachment, we write (A, X) for the argument for X in context A, in other words, for input A the output contains X. Factual detachment is the rule $A \to X/(A, X)$, and says that if there is a rule with the context as antecedent, then the output contains the consequent.

Besides factual detachment, FA systems have the rule of so-called simple aggregation. This one is usually given the form $(A, X), (A, Y)/(A, X \cup Y)$. In this paper aggregation is given the more general form $(A, X), (B, Y)/(A \cup B, X \cup Y)$. This more general form allows for the inputs not to be the same. Given strengthening of the input, $(A, X)/(A \cup B, X)$, the two rules are equivalent. Since we do not assume strengthening of the input, our rule is strictly stronger.

Definition 2.1 [FA system with input] A FA *system* is a triple $\langle L, C, R\rangle$ with L a language, $C \subseteq 2^L \times 2^L$ a set of conditionals written as $A \to X$, and R a set of rules. For every conditional $A \to X \in C$, A and X are finite sets. A FA system is a system $\langle L, C, R\rangle$ where R consists of the rule of *factual detachment* (FD) and the rule of *aggregation* (AND):

$$\text{FD}\ \frac{A \to X}{(A, X)} \qquad \text{AND}\ \frac{(A, X) \quad (B, Y)}{(A \cup B, X \cup Y)}$$

An *input* $I \subseteq L$ for *system* $\langle L, C, R\rangle$ is a subset of the language.

Let $FA = \{FD, AND\}$. We write $a(A \to X) = A$ for the *antecedent* of a conditional, and $c(A \to X) = X$ for the *consequent* of a conditional. We write

$a(C) = \cup\{a(A \to X) \mid A \to X \in C\}$ for the union of the antecedents of all the conditionals in C. We write $c(C) = \cup\{c(A \to X) \mid A \to X \in C\}$ for the union of the consequents of all the conditionals in C. [4]

The following example is meant to exercise the notation. We build a language, and introduce a set of conditionals and an input. The language L is the domain (or universe) of discourse. For the purpose of the example, L is a set of literals. Following Tosatto et al., we also introduce a complement function $\overline{e}$ for the elements e of the language L.

Example 2.2 [Sing and dance, adapted from Goble [3]] Given a language L_0 which does not contain formulas of the form $\sim a$, the language L is $L_0 \cup \{\sim a \mid a \in L_0\}$. For $a \in L$, if $a \in L_0$ then $\overline{a} = \sim a$, and otherwise $\overline{a} = b$ for the $b \in L_0$ such that $a = \sim b$.

Let L_0 be $\{x, y, d, s\}$. Intuitively: "it is Spring" (x); "it is Sunday" (y); "a dance is performed" (d); and "a song is performed" (s). The language L adds classical negation to the language, $L = L_0 \cup \{\sim y, \sim x, \sim d, \sim s\}$. The complement function says $\bar{x} = \sim x$, $\overline{\sim x} = x$, and so on.

Suppose the conditionals $C_1 = \{y \to d, x \to s\}$ apply to a wedding party. This says that on Sundays one ought to dance, and in Spring one ought to sing. The antecedents of the conditionals are: $a(y \to d) = y$; $a(x \to s) = x$; $a(C_1) = \{x, y\}$. Their consequents are: $c(y \to d) = d$; $c(x \to s) = s$; $c(C_1) = \{s, d\}$.

We distinguish three related kinds of output from a system and an input, called derivations, arguments and detachments, respectively. A *derivation* is a finite tree, whose leaves are elements from the set of conditionals and whose root is a pair (A, X) obtained by successive applications of the rules, with the further constraint that $A \subseteq I$. [5] An *argument* is a pair (A, X) for which such a derivation exists, and X is a *detachment* for which such an argument (A, X) exists. [6]

Definition 2.3 [Derivations `der`, Arguments *arg*, and Detachments *det*]
Given a system $\langle L, C, R\rangle$ and an input I,

- a *derivation* of (A, X) on the basis of I in system $\langle L, C\rangle$ is a finite tree [7] using the rules R, with as leaves elements of C, and as root the pair (A, X) where $A \subseteq I$ and $X \subseteq L$.

[4] To ease readability we will omit curly braces when referring to singleton sets, and we write $a \to x$ for $\{a\} \to \{x\}$.

[5] Alternatively, we could add the condition $A \subseteq I$ only to the definitions of arguments and detachments, or only to the definition of detachments. There are pros and cons to both choices. For example, the advantage of our definition is that the set of derivations is smaller, but the disadvantage is that the set of derivations is not closed under sub-derivations, which complicates the proofs of the formal results.

[6] Note the special feature of our formal framework that weakening of the output can be added in different ways. For example, one can add a rule $(A, X \cup Y)/(A, X)$, or one can adapt the definition of detachment such that X is detached for input I if there is an argument (A, Y) such that $A \subseteq I$ and $X \subseteq Y$. The same holds for other properties added to the formal system. We leave the formal analysis of such kinds of extensions to further research.

[7] By a finite tree, we mean one with finitely many nodes.

- an *argument* is a pair (A, X), such that there exists a derivation d with $root(d) = (A, X)$.
- a *detachment* is a set X such that there is an argument (A, X).

We write $\mathtt{der}(L, C, I, R)$ for the set of all the derivations which can be constructed in this way, we write $arg(L, C, I, R)$ for the set of all such arguments, and we write $det(L, C, I, R)$ for the set of all such detachments.

We write $leaves(d)$ for the set of all the leaves of derivation d, $i((A, X)) = A$ for the input of a pair (A, X) and $o((A, X)) = X$ for the output of a pair (A, X). Also we write $i(D) = \cup\{i((A, X)) \mid (A, X) \in D\}$ and $o(D) = \cup\{o((A, X)) \mid (A, X) \in D\}$ for the inputs and outputs of sets of such pairs.

The derivation rules take one datatype, norms, and outputs another, arguments. Nonetheless, the main idea is that derivations are always based on an input. This is reflected by the constraint $i(root(d)) \subseteq I$. But we stress that such a constraint is put on the root of the derivation only, and that all the other nodes need not verify this constraint. Otherwise we would not be able to chain conditionals together. Because of this, the property of closure under sub-derivations does not always hold. It depends on the rules being used. We will see an example of this phenomenon with system FC in Section 3. This also makes the proof of the representation theorem for FC trickier. The standard method of induction over the length of derivations is not available any more.

A derivation is a relative notion, since it is meant to represent the inner structure of an argument. As argued before derivations are tied to the context giving a justification for the argument put forward based on what is, or is not, the case. In the literature, the notion of argument is defined in two ways. Either an argument is viewed as either a pair whose first element is a set of formulas (the support) and second element a formula (the conclusion), or as a derivation in a logical proof system, i.e. a sequence, tree or graph of logical formulas. Here we choose the first definition. In the context of this study, the pair itself denotes a norm. However, it could represent any conditional statement. We use the term argument rather than norm, just to emphasize that we are interested in the relationship between a set of premises and its set of conclusions.

We now can briefly explain the notion of abstraction at stake in the theory of abstract normative systems. Intuitively, the detachment system treats the elements of L as atomic, in the sense that detachments have no relation with the logical structure of language L. Formally, we can replace one language L by another one L', define a one-to-one function f between elements of L and L', and extend f to subsets of L and C. Then we have $f(det(L, C, I, R)) = det(f(L), f(C), f(I), R)$. In this sense, it is an abstract theory.

We continue Example 2.2 to illustrate factual detachment and aggregation, as well as the distinction between derivations, arguments and detachments. In the absence of the rule of strengthening of the antecedent, one cannot derive that X is obligatory in context $A \cup B$ from the fact that X is obligatory in context A. This reflects the idea that arguments are minimal, in the sense that

one cannot add irrelevant elements like B to their support. For example, if the input is $\{A, B\}$ and the sole conditional is $A \to X$, then there is no argument $(A \cup B, X)$. But X will be detached, since the input set triggers the conditional in question. The absence of the rule of strengthening of the antecedent does *not* reflect the fact that rules may leave room for exceptions.

Example 2.4 [Example 2.2 - Continued] Given $L = L_0 \cup \{\sim a \mid a \in L_0\}$, we say that an element $a \in I$ is a violation if there is a detachment containing $\overline{a}$, and this detachment is called a violated obligation. Moreover, we say that a detachment is a cue for action if it is not a violated obligation.

The derivations for $C_1 = \{y \to d, x \to s\}$ and $I_1 = \{x, y\}$ are $\mathtt{der}(L, C_1, I_1, FA) =$

$$\left\{ d_1 = \frac{y \to d}{(y,d)}\ \text{FD}\ ,\ d_2 = \frac{x \to s}{(x,s)}\ \text{FD}\ ,\ d_3 = \frac{\dfrac{x \to d}{(x,d)}\ \text{FD}\quad \dfrac{y \to s}{(y,s)}\ \text{FD}}{(\{x,y\},\{s,d\})}\ \text{AND} \right\},$$

the arguments are $arg(L, C_1, I_1, FA) = \{(y,d), (x,s), (\{x,y\}, \{s,d\})\}$ and the detachments are $det(L, C_1, I_1, FA) = \{\{d\}, \{s\}, \{s,d\}\}$, which are all cues for action. Thus I_1 does not contain violations. Factual detachment derives d and s, and aggregation combines them to $\{d, s\}$. First, note that some strengthening of the input is built in the aggregation inference rule AND, as we derive the conditional norm $(\{x,y\}, \{s,d\})$ whose antecedent is stronger than the antecedent of the conditional norms in C_1. Second, note that, for the context where there is no singing $I_2 = \{x, y, \bar{s}\}$, we obtain exactly the same derivations, arguments and detachments. However, now $\bar{s}$ is a violation, and the detachments $\{s\}$ and $\{s,d\}$ are violated obligations, and only $\{d\}$ is a cue for action.

Now consider $C_2 = \{\{x,y\} \to \{s,d\}\}$ and, e.g., I_2. The derivation is

$$\mathtt{der}(L, C_2, I_2, FA) = \left\{ d_4 = \frac{\{x,y\} \to \{s,d\}}{(\{x,y\},\{s,d\})}\ \text{FD} \right\},$$

the arguments are $arg(L, C_2, I_2, FA) = \{(\{x,y\}, \{s,d\})\}$ and the detachments are $det(L, C_2, I_2, FA) = \{\{s,d\}\}$.

It should not come as a surprise that the set of detachments is syntax-dependent. This follows at once from letting the rule of weakening of the output go. This phenomenon is familiar from the literature on belief revision. [8]

Theorem 2.5 gives a representation result for FA systems. The left-hand side of the bi-conditional pertains to the proof theory, while the right-hand side of it provides a semantic characterization in terms of subset selection. For X to be derivable from a set of conditionals C on the basis of input I, X must be the union of the consequents of finitely many conditionals in C, which are all 'triggered' by the input set I. [9]

[8] For more on the rule of weakening of the output, and the reason why it may be considered counter-intuitive, we refer the reader to the discussion in Goble [3] (see also Parent and van der Torre [7].)

[9] In FA systems, we call 'triggered' those conditionals whose antecedents are in I.

Theorem 2.5 (Representation result, FA) *$X \in det(L, C, I, FA)$ if and only if there is some non-empty and finite $C' \subseteq C$ such that $a(C') \subseteq I$ and $X = c(C')$.*

Proof. See [1]. □

Corollary 2.6 (Monotonicity of *det*) *$det(L, C, I, FA) \subseteq det(L, C', I, FA)$ whenever $C \subseteq C'$.*

The following example illustrates how to calculate the detachments using the semantic characterization described in the statement of Theorem 2.5.

Example 2.7 [Example 2.2 - Continued] We calculate $det(L, C_1, I_1, FA)$, now using Theorem 2.5. The set of conditionals C_1 has three non-empty subsets: $C_{1.1} = \{y \to d\}$, $C_{1.2} = \{x \to s\}$, and $C_{1.3} = \{y \to d, x \to s\}$. Here $a(C_{1.1}) \subseteq I_1$, $a(C_{1.2}) \subseteq I_1$ and $a(C_{1.3}) \subseteq I_1$. Also $c(C_{1.1}) = \{d\}$, $c(C_{1.2}) = \{s\}$ and $c(C_{1.3}) = \{s, d\}$. So $det(L, C_1, I_1, FA) = \{c(C_{1.1}), c(C_{1.2}), c(C_{1.3})\} = \{\{d\}, \{s\}, \{s, d\}\}$.

3 FC systems for cumulative aggregation

In this section we introduce FC systems for cumulative aggregation. FC is much alike FA except that the rule of aggregation AND is replaced with that of cumulative aggregation CAND.

Definition 3.1 [FC system with input] A FC system is a triple $\langle L, C, R\rangle$ where R consists of the following rule of *factual detachment* (FD), and the rule of *cumulative aggregation* (CAND). We write $FC = \{FD, CAND\}$.

$$\text{FD} = \frac{A \to X}{(A, X)} \quad \text{CAND} = \frac{(A, X) \quad (B, Y)}{(A \cup (B \setminus X), X \cup Y)}$$

To illustrate the difference between FA and FC systems, we use the same example as the one that Parent and van der Torre [7] use in order to motivate their rule ACT. We reckon that, compared to the framework described in [7], the present framework does not yield any new insights into the analysis of the example itself.

Example 3.2 [Exercise, from Broome [2]] C contains two conditionals. One says that you ought to exercise hard everyday: $\{\} \to x$. The other says that, if you exercise hard everyday, you ought to eat heartily: $x \to h$. Intuitively, in context $\{\}$, we would like to be able to derive $\{x, h\}$, but not $\{h\}$.

FA systems do not allow us to do it.

Let $I = \{\}$. With simple aggregation the set of derivations is $\texttt{der}(L, C, I, FA) = \left\{d_1 = \frac{\{\} \to x}{(\{\}, x)}\ \text{FD}\right\}$, the set of arguments is $arg(L, C, I, FA) = \{(\{\}, x)\}$ and the set of detachments is $det(L, C, I, FA) = \{\{x\}\}$. Thus the desired obligation is not detached. Norms can be chained together only in so far as the input set contains their antecedent. Let $I' = \{x\}$.

Then the set of derivations is $\mathtt{der}(L, C, I', FA) =$

$$\left\{ d_1 = \frac{\{\} \to x}{(\{\}, x)}\ \text{FD} \quad , \quad d_2 = \frac{x \to h}{(x, h)}\ \text{FD} \quad , \quad d_3 = \frac{\dfrac{\{\} \to x}{(\{\}, x)}\ \text{FD} \quad \dfrac{x \to h}{(x, h)}\ \text{FD}}{(x, \{x, h\})}\ \text{AND} \right\},$$

the set of arguments is $arg(L, C, I', FA) = \{(\{\}, x), (x, h), (x, \{x, h\})\}$ and the detachments are $det(L, C, I', FA) = \{\{x\}, \{h\}, \{x, h\}\}$.

With cumulative aggregation, the derivations for C and $I = \{\}$ are $\mathtt{der}(L, C, I, FC) =$

$$\left\{ d_1 = \frac{\{\} \to x}{(\{\}, x)}\ \text{FD} \quad , \quad d_2 = \frac{\dfrac{\{\} \to x}{(\{\}, x)}\ \text{FD} \quad \dfrac{x \to h}{(x, h)}\ \text{FD}}{(\{\}, \{x, h\})}\ \text{CAND} \right\}$$

The arguments are $arg(L, C, I, FC) = \{(\{\}, x), (\{\}, \{x, h\})\}$ and the detachments are $det(L, C, I, FC) = \{\{x\}, \{x, h\}\}$. Factual detachment allows us to detach $\{x\}$, and cumulative aggregation allows us to detach $\{x, h\}$ in addition. Like in [7], h cannot be derived without x. Intuitively, the obligation to eat heartily no longer holds, if you take no exercise.

Definition 3.3 introduces the functions f and g, to be used later on in the semantic characterization of cumulative aggregation. Intuitively, given a set $D \subseteq L$, the function $f(C, D)$ gathers all the consequents of the conditionals in C that are triggered by D. The function $g(C, I)$ gathers all the sets D that extend the input set I and are closed under $f(C, D)$.

Definition 3.3 [f and g] We define

$$f(C, D) = \bigcup \{X \mid A \to X \in C; A \subseteq D\}$$

$$g(C, I) = \{D \mid I \subseteq D \supseteq f(C, D)\}$$

We illustrate the calculation of functions f and g continuing Example 3.2.

Example 3.4 [Example 3.2 - Continued] Consider the following table. The left-most column shows the relevant subsets C' of C. The middle column shows what consequents can be detached depending on what set D is used as input. The right-most column shows the sets D extending I and closed under $f(C', D)$, for each subset C'.

C'	$f(C', D)$	$g(C', \{\})$
$\{\} \to x$	$\{x\}$	$\{D \mid x \in D\}$
$x \to h$	$\{\}$ if $x \notin D$, $\{h\}$ if $x \in D$	$\{D \mid x \notin D$ or $\{x, h\} \subseteq D\}$
$\{\} \to x$, $x \to h$	$\{x\}$ if $x \notin D$, $\{x, h\}$ if $x \in D$	$\{D \mid \{x, h\} \subseteq D\}$

Theorem 3.5 gives a representation result for FC systems. For X to be derivable from a set of conditionals C on the basis of input I, X must be the union of the consequents of finitely many conditionals in C, which are either

directly triggered by the input set I (in the sense of Footnote 9), or indirectly triggered by the input set I (via a chain of norms).

Theorem 3.5 (Representation result, FC) *$X \in det(L, C, I, FC)$ if and only if there is some non-empty and finite $C' \subseteq C$ such that, for all $D \in g(C', I)$, we have $a(C') \subseteq D$ and $X = f(C', D)$.*

Proof. See [1]. □

We show with an example how to calculate the detachments using the semantic characterization given in the statement of Theorem 3.5.

Example 3.6 [Example 3.4 - Continued] We again calculate $det(L, C, I, FC)$, now using Theorem 3.5. We use the Table shown in Example 3.4.

The top row tells us that, $\{x\} \in det(L, C, I, FC)$. This is because, for all D in $g(C', \{\})$, $f(C', D) = \{x\}$.

The bottom row tells us that, $\{x, h\} \in det(L, C, I, FC)$. This is because, for all D in $g(C', \{\})$, $f(C', D) = \{x, h\}$.

We can also conclude that, $\{h\} \notin det(L, C, I, FC)$ because, for all C', there is a D in $g(C', \{\})$ such that $f(C', D) \neq \{h\}$.

Finally, the set of detachments is $det(L, C, I, FC) = \{\{x\}, \{x, h\}\}$.

4 Some properties of FA systems and FC systems

We start by showing how FA systems and FC systems relate to each other.

Definition 4.1 [Argument subsumption] Argument (A, X) *subsumes* argument (B, Y) if $A \subseteq B$ and $X = Y$. Given two sets of arguments S and T, we say that T subsumes S (notation: $S \sqsubseteq T$), if for all $(B, Y) \in S$ there is an argument $(A, X) \in T$ such that (A, X) subsumes (B, Y).

Example 4.2 Consider the following derivation.

$$d = \frac{(A, X) \quad (A \cup B \cup X, X \cup Y)}{(A \cup B, X \cup Y)} \text{ CAND}$$

The argument $(A \cup B, X \cup Y)$ subsumes the argument $(A \cup B \cup X, X \cup Y)$.

Proposition 4.3 *$arg(L, C, I, FA) \sqsubseteq arg(L, C, I, FC)$.*

Proof. Let $(A, X) \in arg(L, C, I, FA)$, where $A \subseteq I$. Let d be the derivation of (A, X) on the basis of I using the rules FD and AND. Let $leaves(d) = \{A_1 \rightarrow X_1, \ldots, A_n \rightarrow X_n\}$. We have $A = \bigcup_{i=1}^{n} A_i$ and $X = \bigcup_{i=1}^{n} X_i$.[10] That is,

$$(A, X) = (\bigcup_{i=1}^{n} A_i, \bigcup_{i=1}^{n} X_i)$$

One may transform d into a derivation d' of (A', X) on the basis of I using the rules FD and CAND. Keep the leaves and their parent nodes (obtained using

[10] Strictly speaking, this follows from a lemma used in the proof of the representation result for FA systems, Lemma 1 in [1].

FD) as they are in d, and replace any application of AND by an application of CAND. The result will be a tree whose root is

$$(A', X) = (A_1 \cup \bigcup_{i=2}^{n}(A_i \setminus \bigcup_{j=1}^{i-1} X_j), \bigcup_{i=1}^{n} X_i)$$

We have

$$A_1 \cup \bigcup_{i=2}^{n}(A_i \setminus \bigcup_{j=1}^{i-1} X_j) \subseteq \bigcup_{i=1}^{n} A_i \subseteq I \text{ and } \bigcup_{i=1}^{n} X_i = \bigcup_{i=1}^{n} X_i$$

On the one hand, $(A', X) \in arg(L, C, I, FC)$. On the other hand, (A', X) subsumes (A, X). □

Corollary 4.4 *$det(L, C, I, FA) \subseteq det(L, C, I, FC)$*

Proof. This follows at once from Proposition 4.3. □

We now point out a number of other properties of FA and FC systems.

Proposition 4.5 (Applicability) *The rules AND and CAND can be applied to any arguments (A, X) and (B, Y).*

Proof. Trivial. Assume arguments (A, X) and (B, Y). By definition of an argument, $A \subseteq I$, $B \subseteq I$, $X \subseteq L$ and $Y \subseteq L$. Thus, $A \cup B \subseteq I$, $A \cup (B \setminus X) \subseteq I$ and $X \cup Y \subseteq L$. □

Proposition 4.6 (Premises permutation, *FA*) *AND can be applied to two arguments (A, X) and (B, Y) in any order.*

Proof. Straightforward. □

It is noteworthy that Proposition 4.6 fails for CAND, as shown by the following counterexample, where $A \neq B$:

$$\frac{(A, B) \qquad (B, A)}{(A, A \cup B)}\ \text{CAND} \quad \not\Longleftrightarrow \quad \frac{(B, A) \qquad (A, B)}{(B, A \cup B)}\ \text{CAND}$$

The arguments $(A, A \cup B)$ and $(B, A \cup B)$ are distinct.

Proposition 4.7 considers two successive applications of AND, or of CAND.

Proposition 4.7 (Associativity) *Each of AND and CAND is associative, in the sense of being independent of the grouping of the pairs to which it is applied.*

Proof. The argument for AND is straightforward, and is omitted. For CAND, it suffices to show that the pairs appearing at the bottom of the following two derivations are equal:

$$\dfrac{\dfrac{\vdots}{(A, X)} \qquad \dfrac{\dfrac{\vdots}{(B, Y)} \qquad \dfrac{\vdots}{(C, Z)}}{(B \cup (C \setminus Y), Y \cup Z)}}{(A \cup ((B \cup (C \setminus Y)) \setminus X), X \cup Y \cup Z)} \qquad \dfrac{\dfrac{\dfrac{\vdots}{(A, X)} \qquad \dfrac{\vdots}{(B, Y)}}{(A \cup (B \setminus X), X \cup Y)} \qquad \dfrac{\vdots}{(C, Z)}}{(A \cup (B \setminus X) \cup (C \setminus (X \cup Y)), X \cup Y \cup Z)}$$

The fact that the two pairs in question are equal follows at once from the following two laws from set-theory:

$$(A \cup B) \setminus X = (A \setminus X) \cup (B \setminus X) \quad (1)$$
$$B \setminus (X \cup Y) = (B \setminus X) \setminus Y \quad (2)$$

We have:

$$\begin{aligned} A \cup ((B \cup (C \setminus Y)) \setminus X) &= A \cup (B \setminus X) \cup ((C \setminus Y) \setminus X) && \text{[by law (1)]} \\ &= A \cup (B \setminus X) \cup (C \setminus (X \cup Y)) && \text{[by law (2)]} \end{aligned}$$

□

Proposition 4.8 *FA systems are closed under sub-derivations in the following sense: given a derivation $d \in \mathtt{der}(L, C, I, FA)$, for all sub-derivations d' of d, $d' \in \mathtt{der}(L, C, I, FA)$–that is, $i(root(d')) \subseteq I$.*

Proof. Let $d \in \mathtt{der}(L, C, I, FA)$ with $root(d) = (A, X)$ and $A = A_1 \cup \ldots \cup A_n \subseteq I$ and $X = X_1 \cup \ldots \cup X_n$. Without loss of generality, we can assume that $n > 1$. By Proposition 4.7, d can be given the form:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A_1 \to X_1}{(A_1, X_1)}\,\text{FD} \quad \dfrac{A_2 \to X_2}{(A_2, X_2)}\,\text{FD}}{(A_1 \cup A_2, X_1 \cup X_2)}\,\text{AND} \quad \dfrac{\vdots}{(A_3, X_3)}\,\text{FD}}{(A_1 \cup A_2 \cup A_3, X_1 \cup X_2 \cup X_3)}\,\text{AND} \quad \vdots}{\vdots}}{\text{AND}\ \dfrac{(A_1 \cup \ldots \cup A_{n-1}, X_1 \cup \ldots \cup X_{n-1})}{(A_1 \cup \ldots \cup A_n, X_1 \cup \ldots \cup X_n)}} \quad \text{FD}\ \dfrac{A_n \to X_n}{(A_n, X_n)}$$

Let d' be a sub-derivation of d with root (A', X'). Clearly, $A' \subseteq A$, and so $A' \subseteq I$, since $A \subseteq I$. □

Proposition 4.9 *FC systems are not closed under sub-derivations.*

Proof. We prove this proposition by giving a counterexample. Let C be the set of conditionals $\{A \to X, X \to Y\}$ and let $I = \{A\}$. Consider the following derivation:

$$d = \dfrac{\overset{d_1}{\dfrac{\dfrac{A \to X}{(A, X)} \quad \dfrac{X \to Y}{(X, Y)}}{(A, X \cup Y)}} = d_2}{}$$

We have $i(root(d)) \subseteq I$, so that $d \in \mathtt{der}(L, C, I, FC)$. Since $i(root(d_2)) = X$ and $X \not\subseteq I$, $d_2 \notin \mathtt{der}(L, C, I, FC)$. □

Proposition 4.10 (Non-repetition) *For every $d \in \mathtt{der}(L, C, I, FA)$ with root (A, X) and leaves leaves(d), there exists a derivation $d' \in \mathtt{der}(L, C, I, FA)$ with the same root and the same set of leaves, such that each leaf in leaves(d') is used at most once. The same holds for every derivation $d \in \mathtt{der}(L, C, I, FC)$.*

Proof. We only consider the case of FC systems (the argument for FA systems is similar). Assume we have a derivation d with $root(d) = (A, X)$ and $leaves(d) = \{A_1 \to X_1, \ldots A_n \to X_n\}$. By Proposition 4.7, one can transform d into a derivation d' of the form

$$\dfrac{\dfrac{\dfrac{\dfrac{A_1 \to X_1}{(A_1, X_1)}\,\text{FD} \quad \dfrac{A_2 \to X_2}{(A_2, X_2)}\,\text{FD}}{\vdots\ \text{AND}} \quad \dfrac{A_3 \to X_3}{(A_3, X_3)}\,\text{FD}}{\vdots\ \text{AND}} \quad \vdots}{\vdots\ \text{AND}} \quad \vdots \quad \text{FD}\ \dfrac{A_n \to X_n}{(A_n, X_n)}}{(A, X)}$$

Suppose that in d' some $A_l \to X_l$ decorates at least two distinct leaves. We show that we can eliminate the second one. To aid comprehension, let B be mnemonic for the following union, where $l \leq j$:

$$A_1 \cup (A_2 \setminus X_1) \cup (A_3 \setminus (X_1 \cup X_2)) \cup \ldots \cup (A_j \setminus (X_1 \cup \ldots \cup X_{j-1}))$$

Suppose we have the step:

$$\dfrac{\dfrac{\dfrac{\dfrac{A_1 \to X_1}{(A_1, X_1)} \quad \dfrac{A_2 \to X_2}{(A_2, X_2)}}{(A_1 \cup (A_2 \setminus X_1), X_1 \cup X_2)} \quad \dfrac{A_3 \to X_3}{(A_3, X_3)}}{(A_1 \cup (A_2 \setminus X_1) \cup (A_3 \setminus (X_1 \cup X_2)), X_1 \cup X_2 \cup X_3)} \quad \vdots}{(B, \bigcup_{i=1}^{j} X_i)} \quad \dfrac{\dfrac{A_j \to X_j}{(A_j, X_j)} \quad \dfrac{A_l \to X_l}{(A_l, X_l)}}{(B \cup (A_l \setminus \bigcup_{i=1}^{j} X_i), \bigcup_{i=1}^{j} X_i \cup X_l)}$$

where the sub-derivation with root $(B, \bigcup_{i=1}^{j} X_i)$ contains a leaf carrying $A_l \to X_l$. That is, $A_l \to X_l$ is one of $A_1 \to X_1$, ... and $A_j \to X_j$, and it is re-used immediately after $A_j \to X_j$. Since X_l is one of X_1, ... and X_j, $\bigcup_{i=1}^{j} X_i \cup X_l = \bigcup_{i=1}^{j} X_i$. On the other hand, $(A_l \setminus \bigcup_{i=1}^{j} X_i) \subseteq (A_l \setminus \bigcup_{i=1}^{l-1} X_i) \subseteq B$, so that $B \cup (A_l \setminus \bigcup_{i=1}^{j} X_i) = B$. Thus, we can remove from d' all the re-occurrences of the leaves as required.

□

5 Related research

As mentioned in Section 1, the present paper extends the framework described by Tosatto et al. [9] in order to handle conjunction of outputs along with the form of cumulative transitivity introduced by Parent and van der Torre [7].

At the time of writing this paper, we are not able to report any formal result showing how the Tosatto et al. framework relates with the present one. Care should be taken here. On the one hand, the present account does not validate the rule of strengthening of the input, while the Tosatto et al. one does in

the following restricted form: from $(\top, x)$, infer (y, x). On the other hand, in order to relate the proof-theory with the semantics, the authors make a detour through the notion of deontic redundancy [10]. A more detailed comparison between the two accounts is left as a topic for future research.

There are close similarities between the systems described in this paper and the systems of I/O logic introduced by Parent and van der Torre [7]. As explained in the introductory section, our rule CAND is the set-theoretical counterpart of their rule ACT. In both systems, weakening of the output goes away. At the same time there are also important differences between the two settings. First, the present setting remains neutral about the specific language to be used. This one need not be the language of propositional logic. Second, the present account does not validate the rule of strengthening of the input.

Tosatto et al. explain how to instantiate the ANS with propositional logic to obtain fragments of the standard input/output logics [5]. In this section we rerun the same exercise for the systems studied in [7]. Unlike Tosatto et al., we argue semantically, and not proof-theoretically, because of the problem alluded to above: derivations in FC are not closed under sub-derivations.

For the reader's convenience, we first briefly recall the definitions of $\mathcal{O}_1$ and $\mathcal{O}_3$ given by Parent and van der Torre [7]. Given a set X of formulas, and a set N of norms (viewed as pairs of formulas), $N(X)$ denotes the image of N under X, i.e., $N(X) = \{x : (a, x) \in N \text{ for some } a \in X\}$. $Cn(X)$ is the consequence set of X in classical propositional logic. And $x \dashv\vdash y$ is short for $x \vdash y$ and $y \vdash x$. We have $x \in \mathcal{O}_1(N, I)$ whenever there is some finite $M \subseteq N$ such that $M(Cn(I)) \neq \{\}$ and $x \dashv\vdash \wedge M(Cn(I))$. We have $x \in \mathcal{O}_3(N, I)$ if and only if there is some finite $M \subseteq N$ such that $M(Cn(I)) \neq \{\}$ and for all B, if $I \subseteq B = Cn(B) \supseteq M(B)$, then $x \dashv\vdash \wedge M(B)$. [11]

Theorem 5.1 (Instantiation) *Let $\langle L, C, R\rangle$ be a FA system, or a FC system, with L the language of propositional logic (without $\top$) and C a set of conditionals whose antecedents and consequents are singleton sets. Define $N = \{(a, x) \mid \{a\} \to \{x\} \in C\}$. The following applies:*

i) If $X \in det(L, C, I, FA)$, then $\wedge X \in \mathcal{O}_1(N, I)$, where $\wedge X$ is the conjunction of all the elements of X;

ii) If $X \in det(L, C, I, FC)$, then $\wedge X \in \mathcal{O}_3(N, I)$.

Proof. See [1]. □

6 Summary and future work

We have extended the Tosatto et al. framework of abstract normative systems in order to handle conjunction of outputs along with the aggregative form of cumulative transitivity introduced by the last two co-authors of the present

[11] The proof-system corresponding to $\mathcal{O}_1$ has three rules: from (a, x) and $b \vdash a$, infer (b, x) (SI); from (a, x) and (a, y), infer $(a, x \wedge y)$ (AND); from (a, x) and $b \dashv\vdash a$, infer (b, x) (EQ). The proof-system corresponding to $\mathcal{O}_3$ may be obtained by replacing (AND) with (ACT). This is the rule: from (a, x) and $(a \wedge x, y)$, infer $(a, x \wedge y)$.

paper. We have introduced two abstract normative systems, the FA and FC systems. We have illustrated these two systems with examples from literature, and presented two representation theorems for these systems. We have also shown how they relate to the original I/O systems.

FA systems. They supplement factual detachment with the rule of simple aggregation, taking unions of inputs and outputs. The representation theorem shows that the sets of formulas that can be detached in FA precisely correspond to sets of conditionals that generate this output.

FC systems. They supplement factual detachment with the rule of cumulative aggregation, a subtle kind of transitivity or reuse of the output, as introduced in [7]. The representation theorem shows how the cumulative aggregation rule corresponds to the reuse of the detached formulas.

Besides the issues mentioned in the previous section, we are currently investigating the question of how to use FA and FC systems as a basis for a Dung-style argumentation framework.

Acknowledgments We thank three anonymous referees for valuable comments. Leendert van der Torre has received funding from the European Union's H2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No. 690974 for the project "MIREL: MIning and REasoning with Legal texts".

References

[1] Ambrossio, D. A., X. Parent and L. van der Torre, *A representation theorem for abstract cumulative aggregation*, Technical report, University of Luxembourg (2016). URL `http://hdl.handle.net/10993/27364`

[2] Broome, J., "Rationality Through Reasoning," Wiley-Blackwell, 2013.

[3] Goble, L., *A logic of good, should, and would. Part I*, Journal of Philosophical Logic **19** (1990), pp. 169–199.

[4] Koslow, A., "A Structuralist Theory of Logic," Cambridge University Press, Cambridge, 2001.

[5] Makinson, D. and L. van der Torre, *Input-output logics*, Journal of Philosophical Logic **29** (2000), pp. 383–408.

[6] Parent, X. and L. van der Torre, *Aggregative deontic detachment for normative reasoning*, in: *Principles of Knowledge Representation and Reasoning: Proceedings of KR 2014, Vienna, Austria, July 20-24, 2014*.

[7] Parent, X. 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*, Lecture Notes in Computer Science, pp. 149–165.

[8] Ripley, D., *Paradoxes and failures of cut*, Australasian Journal of Philosophy **91** (2013), pp. 139–164.

[9] Tosatto, S., G. Boella, L. van der Torre and S. Villata, *Abstract normative systems*, in: *Principles of Knowledge Representation and Reasoning: Proceedings of KR 2012*, pp. 358–368.

[10] van der Torre, L., *Deontic redundancy*, in: G. Governatori and G. Sartor, editors, *Deontic Logic in Computer Science: Proceedings of DEON 2010* (2010), pp. 11–32.