Christian Straßer
Mathieu Beirlaen
Frederik Van De Putte

Adaptive Logic Characterizations of Input/Output Logic

Abstract. We translate unconstrained and constrained input/output logics as introduced by Makinson and van der Torre to modal logics, using adaptive logics for the constrained case. The resulting reformulation has some additional benefits. First, we obtain a proof-theoretic (dynamic) characterization of input/output logics. Second, we demonstrate that our framework naturally gives rise to useful variants and allows to express important notions that go beyond the expressive means of input/output logics, such as violations and sanctions.

Keywords: Input/output logic, Adaptive logics, Proof theory, Nonmonotonic logic, Deontic logic, Deontic conflicts.

1. Introduction

Input/output-logic. Input/output logic (henceforth I/O logic) was introduced by Makinson and van der Torre [29,30] as a formal tool for modeling non-monotonic reasoning with conditionals. It belongs to a broader family of formal systems developed with this purpose in mind, such as, for instance, Gabbay [17], Crocco et al. [16], Kraus et al. [24], Lehmann et al. [25], and Boutilier [14]. I/O logics also provided the groundworks for Bochman's production inference relations that are useful to model causal and abductive inferences [12]. As argued in [32], the main motivation for I/O logic concerns problems of deontic logic.[1] I/O logics have been used e.g. to model various types of permissions [31,43], to capture the dynamics of normative systems and regulations [13], and to formalize reasoning with contrary-to-duty obligations [32].[2]

Technically speaking, I/O logics (without constraints, cf. infra) are operations that map every pair $\langle \mathcal{G}, \mathcal{A} \rangle$ to an "output", where (i) $\mathcal{G}$ is a set of

[1] See also [20] for a survey of ten such problems, presented using the I/O terminology.

[2] We speak of I/O logic (singular) to denote the overall framework that is common to a number of systems, which we call I/O logic*s*. Alternatively, we will sometimes call the latter I/O operations or I/O functions.

Presented by **Heinrich Wansing**; *Received* December 28, 2015

Studia Logica (2016) 104: 869–916
DOI: 10.1007/s11225-016-9656-1

"input/output pairs" (A, B) where A and B are propositional formulas; (ii) $\mathcal{A}$ stands for "input", i.e. a set of propositional formulas; and (iii) the *output* is also a set of propositional formulas.

In a deontic setting, $\mathcal{A}$ usually represents factual information, $\mathcal{G}$ is a set of conditional obligations (A, B) interpreted e.g. as "A commits you to B" or "Under condition B you are obliged to bring about A",[3] and the output consists of what is obligatory, given the facts at hand. However, as stressed in [29,30], one may also apply I/O logic in other contexts. For instance, in a default logic setting the output can be interpreted as a set of hypotheses, derived from the data $\mathcal{A}$ and a set $\mathcal{G}$ of (normal) defaults.

In either of these interpretations of $\mathcal{A}$ and $\mathcal{G}$, it is often useful to consider a set $\mathcal{C}$ of *constraints* on the output—this notion was introduced in [30]. $\mathcal{C}$ restricts the output in one of two senses, each corresponding to a different style of reasoning. We can require the output as a whole to be consistent with $\mathcal{C}$, or we can impose the weaker requirement that, for each A in the output, $\{A\} \cup \mathcal{C}$ has to be consistent. Depending on the application context $\mathcal{C}$ may represent physical constraints, human rights, etc.

In view of these two styles of reasoning, Makinson and van der Torre define corresponding variants of their I/O logics. Given a set of I/O pairs $\mathcal{G}$, the basic idea is to reason on the basis of maximal subsets of $\mathcal{G}$ that produce an output that is consistent with the constraints in $\mathcal{C}$. See Section 2.3 for the exact definitions and more explanation. We will henceforth speak of *constrained I/O logics* to denote these operations.

This paper. Our aim is to represent I/O logics as deductive systems within a rich modal language.[4] Meta-level expressions such as $A \in \mathcal{A}$ or $(A, B) \in \mathcal{G}$ will be expressible in the object-level, resulting in expressions like $\mathsf{in}A$ and $\mathsf{in}A \Rightarrow \mathsf{out}B$ respectively, where in and out are unary modal operators and $\Rightarrow$ is a binary connective. We will see that this language is not only sufficient to characterize many well-known I/O logics (Sections 2, 3), but it allows us to go beyond the expressive means of I/O logics so as to express useful notions in deontic logic such as violations, sanctions, and permissions (Section 4).

[3] Many subtle distinctions arise when giving meaning to deontic modalities in the context of formal normative systems (e.g., we can read them descriptively or prescriptively, we can read them as ought-to-be or as ought-to-do, etc.): since in this paper these subtleties are not crucial we will not discuss them any further.

[4] There are two representations for I/O logics: a semantic and a syntactic or axiomatic one. In this paper we will focus on I/O logics that have an adequate axiomatic representation. It should be noted that although for many I/O logics presented in the literature there is a known sound and complete axiomatization, there are exceptions such as [43].

In order to characterize constraints, we use a framework for dynamic deductive systems known from non-monotonic logic, namely the framework of adaptive logics (henceforth ALs).[5] First, in Section 2.4, we further extend the modal language—e.g. adding a modal operator con to express constraints. Second, the resulting monotonic modal logic is strengthened by means of a defeasible deductive mechanism, which mimics the selection of maximal sets of I/O pairs at the object-level—see Section 3.1.

Motivation. There are various independent motivations for the formal work presented in this paper. First and foremost, characterizing I/O logics by means of ALs enables us to use dynamic deductive systems for modeling reasoning with conditionals (see Section 3.1). To the best of our knowledge, deductive systems that are adequate with respect to the consequence relations of (constrained) I/O logic have not yet been presented in the literature.

Second, an important advantage of adaptive proofs is that they allow for the representation of various forms of defeasible reasoning. As we argue in Section 3.1, there are two facets of defeasibility both of which are explicated via the dynamics of adaptive proofs. One concerns the non-monotonicity of the consequence relation and is expressed via the notion of final derivability (see Definition 12). Another concerns the more internalised dynamics that occurs in the process of analyzing the given information (without any new input) which is expressed by the notion of derivability-at-a-stage (see Definition 11).

Our third point is related. A stage of an adaptive proof represents a (possibly partial and defeasible) analysis of the given information and can as such be reused and extended in contexts in which we have more input, more rules, or more constraints. This modularity ensures that the reasoning process (explicated by an adaptive proof) need not start from scratch again when the context changes and more analysis is needed. Instead, the dynamic retraction mechanism of adaptive proofs takes care of this.

Fourth, by means of a rich modal language we are able to express inputs, outputs, contraints, and I/O pairs in the object language. As a result, our formalism is more expressive than the original framework of I/O logics (see Section 4) so that it is possible to also express violations, sanctions, and permissions.

---

[5] ALs have been developed for various types of defeasible reasoning in the past. More recently, deontic ALs have been used to model reasoning with conflicting obligations and conditional detachment [10,11,18,33,44,46,50]. The dynamic proof theories of ALs are sound and complete with respect to a selection semantics in the sense of [27]. In this paper, we focus on the proof theory of the adaptive systems.

Fifth, due to its modularity, the modal framework allows for variation in a controlled and systematic way (see Section 5.1). For instance, different types of conflict-handling—so-called "strategies" in the AL terminology—result in alternative, yet equally well-behaved systems, some of which have not yet been defined in the context of I/O logic. As in I/O logic one can easily alter the set of rules that govern the conditionals, adapting it to a given application.

Overview. In Section 2 we present the framework of I/O logic in its unconstrained and constrained form, and we introduce a modal characterization of the unconstrained I/O operations. Next, we extend our modal characterization within the adaptive logics framework so as to obtain non-monotonic modal adaptive systems corresponding to the constrained I/O operations (Section 3). Representation results are provided for our modal characterizations in both the unconstrained setting (Theorem 2) and the constrained setting (Theorem 3).

While the characterization of I/O logics is already possible in a fragment of our modal language, we show in Section 4 that using the full language results in a significant increase of expressive power.

In Section 5 we present some new natural variations on existing I/O logics (Section 5.1), we offer some comparisons to existing systems (Section 5.2), and we make suggestions as to how our framework can be be further extended with priorities, by allowing for quantitative considerations and by using a predicative language (Section 5.3). We conclude the paper in Section 6.

2. I/O Logics

2.1. Unconstrained I/O Logics

Preliminaries Let $\mathcal{W}$ be the set of all propositional formulas of classical logic (**CL**) with the usual connectives $\wedge, \vee, \supset, \neg$ and $\equiv$ based on the propositional letters $a, b, \ldots$. $\mathcal{W}$ also contains the verum constant $\top$ and the falsum constant $\bot$. We will use capital letters $A, B, \ldots$ as meta-variables for propositional formulas.

Where **L** is a logic, Γ is a set of **L**-wffs (well-formed formulas) and A is a wff of **L**, we write $\Gamma \vdash_{\mathbf{L}} A$ ($A \in Cn_{\mathbf{L}}(\Gamma)$) to denote that A is **L**-derivable from Γ, and $\vdash_{\mathbf{L}} A$ to denote that A is **L**-derivable from the empty premise set.

Characterizing the I/O logics. In this section, we define the unconstrained I/O logics from [29]. These are defined as functions that map each pair $\langle \mathcal{G}, \mathcal{A} \rangle$ with a set of facts $\mathcal{A} \subseteq \mathcal{W}$ and a set of I/O pairs $\mathcal{G} \subseteq \mathcal{W} \times \mathcal{W}$ to an output set $\mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) \subseteq \mathcal{W}$, where R is a set of rules for I/O pairs. We sometimes refer to $\mathcal{G}$ as the set of generators. As shown in [29], each of these functions can be characterized in two equivalent ways—one is called "semantic", the other "syntactic". In the remainder of this section we define the I/O functions along the lines of the syntactic characterization.[6]

Some candidate rules are:

$$\text{If } A \vdash_{\mathbf{CL}} B \text{ and } (B, C), \text{ then } (A, C). \tag{SI}$$

$$\text{If } A \vdash_{\mathbf{CL}} B \text{ and } (C, A), \text{ then } (C, B). \tag{WO}$$

$$\text{If } (A, B) \text{ and } (A, C), \text{ then } (A, B \wedge C). \tag{AND}$$

$$\text{If } (A, C) \text{ and } (B, C), \text{ then } (A \vee B, C). \tag{OR}$$

$$\text{If } (A, B) \text{ and } (A \wedge B, C), \text{ then } (A, C). \tag{CT}$$

$$(A, A). \tag{ID}$$

$$(\top, \top) \tag{Z}$$

$$(\bot, \bot) \tag{F}$$

$$\text{If } A \vdash_{\mathbf{CL}} B,\ B \vdash_{\mathbf{CL}} A \text{ and } (C, A), \text{ then } (C, B). \tag{EQ}$$

$$\text{If } (A, B) \text{ and } (A \wedge B, C), \text{ then } (A, B \wedge C). \tag{ACT}$$

We will sometimes write $\hat{\mathsf{R}}$ denoting the set of rules in R together with all the rules that are derivable from R. For instance, where R contains (WO), (EQ) $\in \hat{\mathsf{R}}$.[7]

In this paper we will consider sets of rules R containing at least (SI), (AND) and (EQ) (i.e., (SI), (AND), (EQ) $\in \hat{\mathsf{R}}$). We shall refer to such sets as *normal sets of rules*. All the sets of rules originally defined in [29] are normal (they consist of at least (SI), (AND) and (WO) which makes (EQ) derivable). The *production inference relations* in [12] validate additionally

[6] Often the syntactic versions of I/O logics are written as $\mathrm{deriv}_{\mathsf{R}}$ while $\mathrm{out}_{\mathsf{R}}$ is reserved for the semantic versions. In this paper we will stick to $\mathrm{out}_{\mathsf{R}}$ for the syntactic versions due to its suggestive name: the function produces output.

[7] More precisely, $\hat{\mathsf{R}}$ is the maximal superset of R such that for all sets of I/O-pairs $\mathcal{G}$, applying R to $\mathcal{G}$ results in the same set of I/O-pairs as applying $\hat{\mathsf{R}}$ to $\mathcal{G}$.

Table 1. Some I/O functions in terms of the rules they use. The asterisk indicates derivable rules

	(Z)	(SI)	(WO)	(AND)	(OR)	(CT)	(ID)	(EQ)	(ACT)
out_1	✓	✓	✓	✓				✓*	
$\mathrm{out}_{1'}$		✓		✓				✓	
out_2	✓	✓	✓	✓	✓			✓*	
$\mathrm{out}_{2'}$		✓		✓	✓			✓	
out_3	✓	✓	✓	✓		✓		✓*	✓*
$\mathrm{out}_{3'}$		✓		✓				✓	✓
out_4	✓	✓	✓	✓	✓	✓		✓*	✓*
out_1^+	✓*	✓	✓	✓			✓	✓*	
out_2^+	✓*	✓	✓	✓	✓	✓*	✓	✓*	✓*
out_3^+	✓*	✓	✓	✓		✓	✓	✓*	✓*
out_4^+	✓*	✓	✓	✓	✓	✓	✓	✓*	✓*

the rule (F). Sets of rules without (WO) were defined in [35,36] which extend on results in [41,42].[8]

By $\mathcal{G}_{\mathsf{R}}$ we denote the closure of $\mathcal{G}$ under R.[9] The general form of the syntactic construction is given by Definition 1.

Definition 1. $B \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ iff there are $A_1, \ldots, A_n \in \mathcal{A}$ such that $\left(\bigwedge_1^n A_i, B\right) \in \mathcal{G}_{\mathsf{R}}$.

Fact 1. $\mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) =_{\mathsf{df}} \left\{B \mid \textit{ for some } A \in \mathrm{Cn}_{\mathbf{CL}}(\mathcal{A}), (A, B) \in \mathcal{G}_{\mathsf{R}}\right\}$.[10]

Table 1 shows, among others, how the I/O operations out_1 to out_4 and out_1^+ to out_4^+ from [29] are obtained by combinations of the rules defined above.[11] Note that all of them make use of (SI), (AND) and (WO). To cover the border case where $\mathcal{G} = \emptyset$, one needs to add the (zero premise) rule (Z) in order to ensure that all tautologies are in the output (see [30, p. 157]). The sets of rules $\mathrm{out}_{1'}, \mathrm{out}_{2'}$ and $\mathrm{out}_{3'}$ have been defined in [35,36]. The main difference with respect to the rules from [29] is that these rules give up on (WO) and instead only make use of the weaker (EQ).

[8] More precisely, the reconstructions we offer in subsequent sections will be adequate for normal sets of rules given that the set of facts $\mathcal{A}$ is **CL**-consistent or that (F) $\in \hat{\mathsf{R}}$. Our reconstructions trivialize inconsistent sets of facts while the corresponding original I/O logics do not trivialize them, except when rules such as (F) are derivable.

[9] As usual, the closure of a set $\mathcal{X}$ under a set of rules R is the smallest superset of $\mathcal{X}$ that is closed under applications of rules in R.

[10] Here is why: ($\Rightarrow$) is trivial. ($\Leftarrow$) By **CL**-properties there are $A_1, \ldots, A_n \in \mathcal{A}$ such that $\bigwedge_1^n A_i \vdash A$. By (SI), $\left(\bigwedge_1^n A_i, B\right) \in \mathcal{G}_{\mathsf{R}}$.

[11] out_2^+ and out_4^+ coincide. This was first noted in [30].

When giving examples in this paper, we will only use the operation of *simple-minded output* out_1 which allows us to focus on the formal novelties that are introduced in the current paper. For an elaborate discussion of the other I/O functions, the reader is referred to [29,30,32,35,36,41,42].

Let us note some general properties of the I/O functions. The first two facts show that Definition 1 warrants syntax-independency:

FACT 2. *Where* $Cn_{\mathbf{CL}}(\mathcal{A}) = Cn_{\mathbf{CL}}(\mathcal{A}')$, $\mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) = \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}')$.

FACT 3. *Where* $\mathcal{G}_{\mathsf{R}} = \mathcal{G}'_{\mathsf{R}}$, $\mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) = \mathrm{out}_{\mathsf{R}}(\mathcal{G}', \mathcal{A})$.

The output of normal sets of rules is closed under conjunction and under classical equivalence:

THEOREM 1. *1. If* $A, B \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ *then* $A \wedge B \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$.

2. If $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$, $A \vdash_{\mathbf{CL}} B$, *and* $B \vdash_{\mathbf{CL}} A$, *then* $B \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$.

PROOF. Ad 1: Suppose $A, B \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. Hence, there are $C, D \in \mathrm{Cn}_{\mathbf{CL}}(\mathcal{A})$ for which $(C, A), (D, B) \in \mathcal{G}_{\mathsf{R}}$. By (SI), also $(C \wedge D, A), (C \wedge D, B) \in \mathcal{G}_{\mathsf{R}}$. By (AND), $(C \wedge D, A \wedge B) \in \mathcal{G}_{\mathsf{R}}$. Hence, also $A \wedge B \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. Ad 2: Suppose the antecedent holds. Thus, there is a $C \in \mathrm{Cn}_{\mathbf{CL}}(\mathcal{A})$ for which $(C, A) \in \mathcal{G}_{\mathsf{R}}$. By (EQ), $(C, B) \in \mathcal{G}_{\mathsf{R}}$ and thus $B \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. ■

Example 1 illustrates the interplay between input, generators and output for the specific case of out_1.

EXAMPLE 1. Let $\mathcal{A} = \{a, b\}, \mathcal{G} = \{(a, c), (b, d), (a \wedge b, e)\}$. By (SI) we can derive $(a \wedge b, c)$ and $(a \wedge b, d)$ from (a, c) and (b, d) respectively. By (AND), we obtain $(a \wedge b, c \wedge d)$. Since $a \wedge b \in Cn_{\mathbf{CL}}(\mathcal{A})$, $(c \wedge d) \in \mathrm{out}_1(\mathcal{G}, \mathcal{A})$. Similarly, we get $e \in \mathrm{out}_1(\mathcal{G}, \mathcal{A})$ since $a \wedge b \in Cn_{\mathbf{CL}}(\mathcal{A})$ and $(a \wedge b, e) \in \mathcal{G}$.

2.2. An Alternative Characterization

In this section, we provide a modal logic characterization of the I/O operations from the preceding section.[12] As we will argue in Section 3, the

[12] In [29] Makinson and van der Torre also present modal characterizations of some (unconstrained) I/O functions. However, their translation does not cover the four cases where (OR) is invalid. Moreover, it is hard to see how this translation can be adjusted to the context of constrained I/O logics. For this reason, we present a different modal characterization. We return to this point in Section 5.1. Bochman in [12] presents another semantic characterization of his 'production inference relations' (i.e., I/O logics that satisfy (SI), (WO), (AND), (Z) and (F)) based on classical bimodels (i.e., pairs of **CL**-consistent and **CL**-deductively closed sets).

resulting systems allow us to model reasoning about inputs, conditionals and output in a natural and very expressive language.

Where, as before, $\mathcal{W}$ is the set of well-formed formulas of **CL**, the set $\mathcal{W}'$ of well-formed formulas is given by the following grammar:

$$\begin{aligned}\mathcal{W}' := \langle\mathcal{W}\rangle \mid \mathsf{in}\langle\mathcal{W}\rangle \mid \mathsf{out}\langle\mathcal{W}\rangle \mid \langle\mathcal{W}'\rangle \Rightarrow \langle\mathcal{W}'\rangle \mid \neg\langle\mathcal{W}'\rangle \mid \langle\mathcal{W}'\rangle \vee \langle\mathcal{W}'\rangle \mid \\ \langle\mathcal{W}'\rangle \wedge \langle\mathcal{W}'\rangle \mid \langle\mathcal{W}'\rangle \supset \langle\mathcal{W}'\rangle \mid \langle\mathcal{W}'\rangle \equiv \langle\mathcal{W}'\rangle\end{aligned}$$

Given a pair $\langle\mathcal{G}, \mathcal{A}\rangle$, the idea is to represent 'factual' inputs $A \in \mathcal{A}$ by $\mathsf{in}A$, I/O pairs (A, B) in $\mathcal{G}$ by $\mathsf{in}\,A \Rightarrow \mathsf{out}\,B$, and outputs by $\mathsf{out}B$. Let us henceforth abbreviate $\mathsf{in}\,A \Rightarrow \mathsf{out}\,B$ by $A \to B$.

Definition 2. $\Gamma^{\mathcal{G},\mathcal{A}} = \{A \to B \mid (A, B) \in \mathcal{G}\} \cup \{\mathsf{in}A \mid A \in \mathcal{A}\}$

We interpret the input operator in as a **KD**-modality:

$$\vdash \mathsf{in}(A \supset B) \supset (\mathsf{in}A \supset \mathsf{in}B) \tag{Kin}$$

$$\vdash \mathsf{in}A \supset \neg\mathsf{in}\neg A \tag{Din}$$

$$\text{If } \vdash A \text{ then } \vdash \mathsf{in}A \tag{NECin}$$

Note that in view of I/O logics that do not validate (WO) it would be too strong to also model out as a **K**-modality. For such logics we can simply let out be a property-less dummy-operator. For the characterization of other I/O logics we can model out as a **K**-modality.[13]

The binary connective $\Rightarrow$ is fully characterized by modus ponens:

$$A, A \Rightarrow B \vdash B \tag{MP$_\Rightarrow$}$$

Note that modus ponens for $\Rightarrow$ allows to derive $\mathsf{out}B$ from $\mathsf{in}A$ and $A \to B$. We write (MP$_\supset$) for modus ponens relative to $\supset$.

Where R is given, let $\mathsf{R}^\to$ denote the associated set of rules for conditionals in which each rule in R (where $A_i, B_i, C_j, D_j, E, F \in \mathcal{W}$) is translated according to the translation schemes in Table 2.[14]

For instance, (CT) is translated to

$$\vdash ((A \to B) \wedge ((A \wedge B) \to C)) \supset (A \to C)$$

[13]It should be noted that our representation theorems do not depend on the fact that out is interpreted as a **K**-modality. In other words, this interpretation is *admissible* as soon as the corresponding I/O logic validates (WO).

[14]We opted for a conditional $\Rightarrow$ that is weaker than material implication $\supset$ since we want to also characterise rather weak I/O logics. For instance, were we to choose $\supset$ as our $\Rightarrow$, OR$^\to$ would be a derived rule.

Table 2. Translation schemes

Rule in R	Translated rule in $\mathsf{R}^{\rightarrow}$
If $A_1 \vdash_{\mathbf{CL}} B_1, \ldots, A_n \vdash_{\mathbf{CL}} B_n$ and $(C_1, D_1), \ldots, (C_m, D_m)$ then (E, F).	If $A_1 \vdash B_1, \ldots, A_n \vdash B_n$ then $\vdash (C_1 \rightarrow D_1) \wedge \ldots \wedge (C_m \rightarrow D_m) \supset (E \rightarrow F)$.
If $(C_1, D_1), \ldots, (C_m, D_m)$ then (E, F).	$\vdash ((C_1 \rightarrow D_1) \wedge \ldots \wedge (C_m \rightarrow D_m)) \supset (E \rightarrow F)$

and (WO) is translated to

If $A \vdash B$ then $\vdash (C \rightarrow A) \supset (C \rightarrow B)$.

Definition 3. (*Modal I/O logics*) The logic $\mathbf{MIO}^{-}_{\mathsf{R}}$ is defined by adding (Kin) and (Din) to the axioms of $\mathbf{CL}$, and by closing the resulting set under (NECin), the rules in $\mathsf{R}^{\rightarrow}$, ($\mathrm{MP}_{\supset}$), and ($\mathrm{MP}_{\Rightarrow}$). $\mathbf{MIO}^{+}_{\mathsf{R}}$ is defined analogously, just that $\mathtt{out}$ is a $\mathbf{K}$-modality.

We write $\mathbf{MIO}_{\mathsf{R}}$ whenever we refer to any of the two variants.

Definition 4. Where A is in $\mathcal{W}'$ and $\Gamma \subseteq \mathcal{W}'$, we write $\Gamma \vdash_{\mathbf{MIO}_{\mathsf{R}}} A$ to denote that A is $\mathbf{MIO}_{\mathsf{R}}$-derivable from Γ.

This completes our modal characterization of the I/O functions, resulting in the following representation theorem[15]:

Theorem 2. *Where* $\mathcal{A}$ *is* $\mathbf{CL}$*-consistent or* (F) $\in \hat{\mathsf{R}}$*:*

1. $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ *iff* $\Gamma^{\mathcal{G},\mathcal{A}} \vdash_{\mathbf{MIO}^{-}_{\mathsf{R}}} \mathsf{out}A$
2. *where* (WO), (Z) $\in \hat{\mathsf{R}}$, $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ *iff* $\Gamma^{\mathcal{G},\mathcal{A}} \vdash_{\mathbf{MIO}^{+}_{\mathsf{R}}} \mathsf{out}A$.

Note that our representation theorem does not cover the border case where $\mathcal{A}$ is inconsistent. In this case, it follows from the $\mathbf{KD}$-properties of in that $\Gamma^{\mathcal{G},\mathcal{A}} \vdash_{\mathbf{MIO}_{\mathsf{R}}} \bot$ and hence also $\Gamma^{\mathcal{G},\mathcal{A}} \vdash_{\mathbf{MIO}_{\mathsf{R}}} \mathsf{out}A$ for all $A \in \mathcal{W}$. On the other hand, where $\mathcal{A}$ is inconsistent, $\mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ need not be trivial in case (F) is not derivable from R. In that case: $\mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) = \{B \mid (A, B) \in \mathcal{G}_{\mathsf{R}}\}$.

In the remainder, let $\mathbf{MIO_1}$ denote the modal logic that corresponds to the operation of simple-minded output out_1 defined in the previous section. We briefly illustrate $\mathbf{MIO_1}$ by means of our previous example. Recall, $\mathcal{A} = \{a, b\}$ and $\mathcal{G} = \{(a, c), (b, d), (a \wedge b, e)\}$. Applying Definition 2, we obtain the premise set $\Gamma_1 = \{\mathsf{in}a, \mathsf{in}b, a \rightarrow c, b \rightarrow d, (a \wedge b) \rightarrow e\}$. By the $\mathbf{KD}$-properties of in, we can derive $\mathsf{in}(a \wedge b)$ from the first two premises. By the rule (SI),

[15]See Appendix 1 for the proof of Theorem 2.

we can derive $(a \wedge b) \to c$ and $(a \wedge b) \to d$ from the third and fourth premise respectively. Applying (AND), we obtain $(a \wedge b) \to (c \wedge d)$. Finally, by ($\mathrm{MP}_{\Rightarrow}$), we can derive $\mathsf{out}(c \wedge d)$ from $\mathsf{in}(a \wedge b)$ and $(a \wedge b) \to (c \wedge d)$. Also, from $\mathsf{in}(a \wedge b)$ and the premise $(a \wedge b) \to e$, we can derive $\mathsf{out}e$ by means of ($\mathrm{MP}_{\Rightarrow}$).

2.3. Constrained I/O Logics

In [30], Makinson and van der Torre extend their I/O framework in order to deal with excess output. Such an excess can arise in various ways. The output may be inconsistent per se, or the output may be inconsistent with the input. Suppose, for instance, that $\mathcal{G} = \{(\top, \neg a), (a, b)\}$ and $\mathcal{A} = \{a\}$. Then $\mathrm{out}_1(\mathcal{G}, \mathcal{A}) = \mathrm{Cn}(\{\neg a, b\})$ is consistent, but inconsistent with the input a. For the operations that use the rule (ID), both types of excess coincide.

More generally, one may also think of excessive output as output which conflicts with certain physical, practical or normative principles. To cover all such cases, a *constraint set* $\mathcal{C} \subseteq \mathcal{W}$ is introduced. The cases $\mathcal{C} = \emptyset$ and $\mathcal{C} = \mathcal{A}$ allow us to express consistency of output, and its consistency with input $\mathcal{A}$ respectively.

The strategy used by Makinson and van der Torre for eliminating excess output is "to cut back the set of generators to just below the threshold of yielding excess" [30, p. 160]. Using well-known techniques from non-monotonic logic, they prune the set of generators to obtain its maximal non-excessive subsets.

Definition 5. $\mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})$ is the family of all maximal $\mathcal{H} \subseteq \mathcal{G}$ such that $\mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A})$ is consistent with $\mathcal{C}$.

Remark 1. $\mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A})$ is inconsistent with $\mathcal{C}$ iff $\mathcal{C}' \vdash_{\mathbf{CL}} A$ for some finite $\mathcal{C}' \subseteq \mathcal{C}$ and where $\neg A \in \mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A})$.[16]

Similar to the notions of skeptical resp. credulous consequence from non-monotonic logic, Definition 5 gives rise to operations of full meet resp. full join constrained output.[17]

Definition 6. (*Full meet constrained output*)

$$\mathrm{out}^{\cap}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C}) =_{\mathrm{df}} \bigcap\{\mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A}) \mid \mathcal{H} \in \mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})\}$$

[16] Here's why: $\mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A}) \cup \mathcal{C} \vdash_{\mathbf{CL}} \bot$ iff [by the compactness of **CL**] $\mathcal{C}' \cup \mathcal{O} \vdash_{\mathbf{CL}} \bot$ where $\mathcal{C}'$ is a finite subset of $\mathcal{C}$ and $\mathcal{O}$ is a finite subset of $\mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A})$ iff $\mathcal{C}' \vdash_{\mathbf{CL}} \neg \bigwedge \mathcal{O}$. The rest follows immediately since by Theorem 1 $\neg\neg \bigwedge \mathcal{O} \in \mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A})$.

[17] For readers unfamiliar with the notions of skeptical and credulous consequence, we refer to [22,28] for more details.

Definition 7. (*Full join constrained output*)

$$\mathrm{out}^{\cup}_{\mathsf{R}}(\mathcal{G},\mathcal{A},\mathcal{C}) =_{\mathrm{df}} \bigcup\{\mathrm{out}_{\mathsf{R}}(\mathcal{H},\mathcal{A}) \mid \mathcal{H} \in \mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G},\mathcal{A},\mathcal{C})\}$$

Conventionally we set $\mathrm{out}^{\cap}_{\mathsf{R}}(\mathcal{G},\mathcal{A},\mathcal{C}) = \mathcal{W}$ and $\mathrm{out}^{\cup}_{\mathsf{R}}(\mathcal{G},\mathcal{A},\mathcal{C}) = \mathcal{W}$ for the border case in which $\mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G},\mathcal{A},\mathcal{C}) = \emptyset$. Note that this is exactly the case in which $\mathrm{out}_{\mathsf{R}}(\emptyset,\mathcal{A}) \cup \mathcal{C}$ is not consistent.

As mentioned in the introduction, there are two ways in which one may pose constraints on the output. According to the first, the output set as a whole is required to be consistent with $\mathcal{C}$; according to the second, each formula A in the output is required to be consistent with $\mathcal{C}$. Definition 6 gives us an operation that respects the first requirement, whereas Definition 7 results in an operation that respects the second.

Example 2. Let again $\mathcal{G} = \{(a,c),(b,d),(a \wedge b,e)\}$ and $\mathcal{A} = \{a,b\}$. We add the constraint set $\mathcal{C} = \{\neg(c \wedge d)\}$. We have: $\mathsf{maxfamily}_1(\mathcal{G},\mathcal{A},\mathcal{C}) = \{\mathcal{H},\mathcal{H}'\}$, where $\mathcal{H} = \{(a,c),(a \wedge b,e)\}$ and $\mathcal{H}' = \{(b,d),(a \wedge b,e)\}$. By Definition 6, $c \notin \mathrm{out}^{\cap}_1(\mathcal{G},\mathcal{A},\mathcal{C})$ and $d \notin \mathrm{out}^{\cap}_1(\mathcal{G},\mathcal{A},\mathcal{C})$. However, $c \vee d \in \mathrm{out}^{\cap}_1(\mathcal{G},\mathcal{A},\mathcal{C})$. To see why, note that by (WO), $(a, c \vee d) \in \mathcal{H}_{\mathsf{R}}$ and $(b, c \vee d) \in \mathcal{H}'_{\mathsf{R}}$. Hence $c \vee d \in \mathrm{out}_1(\mathcal{H},\mathcal{A}) \cap \mathrm{out}_1(\mathcal{H}',\mathcal{A})$.

By Definition 7, $c \in \mathrm{out}^{\cup}_1(\mathcal{G},\mathcal{A},\mathcal{C})$ and $d \in \mathrm{out}^{\cup}_1(\mathcal{G},\mathcal{A},\mathcal{C})$. However, $c \wedge d \notin \mathrm{out}^{\cup}_1(\mathcal{G},\mathcal{A},\mathcal{C})$.

In [30, Section 6] Makinson and van der Torre investigate constrained I/O logics in terms of so-called constrained derivations. This way, they obtain a syntactic characterization of the full join constrained I/O operations based on $\mathrm{out}_1,\ldots,\mathrm{out}_4$ and $\mathrm{out}^+_1,\ldots,\mathrm{out}^+_4$ for the specific case where $\mathcal{C} = \mathcal{A} = \{A\}$ for some $A \in \mathcal{W}$ (see [30, Observation 9]). In Sections 3.1 and 3.2, we provide a proof theory for *all* constrained I/O operations, for the general case of arbitrary $\mathcal{A} \subseteq \mathcal{W}$ and $\mathcal{C} \subseteq \mathcal{W}$.

2.4. Extending the Alternative Characterization with Constraints

Before we can provide proof theories for the constrained I/O operations, we need to extend our modal characterization from Section 2.2. We define a new language $\mathcal{W}^c$ which enriches $\mathcal{W}'$ in two ways. First we add a modal operator con for modeling constraints. Second, we add a unary operator $\bullet$. In the characterization of I/O logic we will use it for prefixing I/O pairs in the set $\mathcal{G}$ (the role of the $\bullet$-operator is addressed below).

$$\begin{aligned}\mathcal{W}^c := {} & \langle\mathcal{W}\rangle \mid \mathsf{in}\langle\mathcal{W}\rangle \mid \mathsf{out}\langle\mathcal{W}\rangle \mid \mathsf{con}\langle\mathcal{W}\rangle \mid \langle\mathcal{W}^c\rangle \Rightarrow \langle\mathcal{W}^c\rangle \mid \bullet\langle\mathcal{W}^c\rangle \mid \\ & \neg\langle\mathcal{W}^c\rangle \mid \langle\mathcal{W}^c\rangle \vee \langle\mathcal{W}^c\rangle \mid \langle\mathcal{W}^c\rangle \wedge \langle\mathcal{W}^c\rangle \mid \langle\mathcal{W}^c\rangle \supset \langle\mathcal{W}^c\rangle \mid \langle\mathcal{W}^c\rangle \equiv \langle\mathcal{W}^c\rangle\end{aligned}$$

The modal operator con is characterized as a **KD**-modality:

$$\vdash \mathsf{con}(A \supset B) \supset (\mathsf{con}A \supset \mathsf{con}B) \tag{Kcon}$$

$$\vdash \mathsf{con}A \supset \neg\mathsf{con}\neg A \tag{Dcon}$$

$$\text{If } \vdash A \text{ then } \vdash \mathsf{con}A \tag{NECcon}$$

Moreover, we add an axiom schema expressing that outputs should respect constraints:

$$\vdash \mathsf{con}A \supset \neg\mathsf{out}\neg A \tag{ROC}$$

DEFINITION 8. $\mathbf{CMIO_R}^-$ is obtained by adding (Kcon), (Dcon), and (ROC) to the axioms of $\mathbf{MIO_R}^-$ and by closing the resulting set under (NECin) (NECout), (NECcon), $\mathsf{R}^{\rightarrow}$, (MP$_\supset$) and (MP$_\Rightarrow$). $\mathbf{CMIO_R}^+$ is defined analogously, just that out is a **K**-modality.

We write $\mathbf{CMIO_R}$ whenever we refer to any of the two variants.

As mentioned above, Makinson and van der Torre deal with constraints by pruning the generating set $\mathcal{G}$. Whereas in the constrained I/O systems I/O pairs in $\mathcal{G}$ are selected, in our system conditionals are *activated* by removing the bullet. First, I/O pairs in $\mathcal{G}$ are prefixed with the $\bullet$-operator, which functions as a dummy operator. Next, a *non-activated* conditional $\bullet(A \rightarrow B)$ is *activated* by inferring from it the conditional $A \rightarrow B$. Once activated, we can detach the output of triggered conditionals by means of (MP$_\Rightarrow$).

Definition 2 is adjusted accordingly:

DEFINITION 9. $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} =_{\mathsf{df}} \{\bullet(A \rightarrow B) \mid (A, B) \in \mathcal{G}\} \cup \{\mathsf{in}A \mid A \in \mathcal{A}\} \cup \{\mathsf{con}A \mid A \in \mathcal{C}\}$

For instance, where as before, $\mathcal{G} = \{(a, c), (b, d), (a \wedge b, e)\}$, $\mathcal{A} = \{a, b\}$ and $\mathcal{C} = \{\neg(c \wedge d)\}$, this is translated into the premise set $\{\bullet(a \rightarrow c), \bullet(b \rightarrow d), \bullet((a \wedge b) \rightarrow e), \mathsf{in}a, \mathsf{in}b, \mathsf{con}\neg(c \wedge d)\}$.

The systems characterized in Definition 8 are not yet equipped with a rule for activating conditionals. We cannot simply add the rule "If $\bullet(A \rightarrow B)$, then $A \rightarrow B$". Instead, we need a logic that can distinguish between cases in which activating a conditional is sensible and cases in which it is not. For instance, given the set $\{\mathsf{in}a, \mathsf{in}b, \bullet(a \rightarrow c), \bullet(b \rightarrow d), \mathsf{con}\neg d\}$, we want to be able to derive $a \rightarrow c$ from $\bullet(a \rightarrow c)$ so that $\mathsf{out}c$ is derivable by means of (MP$_\Rightarrow$). However, given the constraint $\neg d$ we also want to *block* the derivation of $b \rightarrow d$ from $\bullet(b \rightarrow d)$, otherwise we could again apply (MP$_\Rightarrow$) in order to derive $\mathsf{out}d$.

What we are looking for, then, is a *defeasible* mechanism for strengthening the logics from Definition 8. To simplify slightly, we are looking for a mechanism that enables us to infer activated conditionals $A \rightarrow B$ from non-activated conditionals $\bullet(A \rightarrow B)$, in such a way that B is in the output of $\mathcal{G}$ given $\mathcal{A}$ whenever $A \in \mathcal{A}$ and $(A, B) \in \mathcal{G}$ *unless* e.g. there is a constraint preventing the derivation of B.[18] In what follows, we define such a mechanism which is rich enough to characterize, for instance, all the I/O functions defined in Section 2.3.

3. Adaptive Logic Characterizations of I/O Logics

3.1. Dynamic Proofs for I/O Logics

Adaptive logics. The proof theory to be provided for the constrained I/O operations from Section 2.3 is that of the adaptive logics framework (see e.g. [6, 45] for a general introduction). An AL is usually characterized as a triple:

(a) ALs are built 'on top' of a *lower limit logic* (LLL). The AL allows for the application of all inferences valid in the LLL. The LLL has to be monotonic, transitive, reflexive and compact.[19]

(b) ALs strengthen their LLL by considering a set of formulas as false 'as much as possible'. This set of formulas is called the *set of abnormalities* and is denoted by Ω. The members of the set of abnormalities are required to be of a specific logical form, which depends on the application. For instance, in the setting of inconsistency-tolerant (paraconsistent) logic, the set may contain all formulas of the logical form $A \wedge \neg A$ (see e.g. [3]).

(c) The phrase 'as much as possible' in (b) is disambiguated by an *adaptive strategy*. The strategy specifies how to proceed in cases where e.g. we know that at least one of a number of abnormalities is true, but we do not know which. In such cases, reasoners may proceed in various ways: some more, some less cautious. We introduce various such strategies below.

---

[18] The situation is more complex e.g. in cases where $(A, B) \in \mathcal{G}_{\mathsf{R}} \setminus \mathcal{G}$. We explain the technical details in Section 3.

[19] Where $\mathbf{L}$ is a logic, Γ, Γ' are sets of $\mathbf{L}$-wffs, and A is an $\mathbf{L}$-wff: $\mathbf{L}$ is reflexive iff, for all Γ, $\Gamma \subseteq \mathrm{Cn}_{\mathbf{L}}(\Gamma)$; it is transitive iff, for all Γ and Γ', if $\Gamma' \subseteq \mathrm{Cn}_{\mathbf{L}}(\Gamma)$ then $\mathrm{Cn}_{\mathbf{L}}(\Gamma \cup \Gamma') \subseteq \mathrm{Cn}_{\mathbf{L}}(\Gamma)$; it is monotonic iff, for all Γ and Γ', $\mathrm{Cn}_{\mathbf{L}}(\Gamma) \subseteq \mathrm{Cn}_{\mathbf{L}}(\Gamma \cup \Gamma')$; and it is compact iff, for all Γ, Γ', A, if $\Gamma \vdash_{\mathbf{L}} A$, then $\Gamma' \vdash_{\mathbf{L}} A$ for some finite $\Gamma' \subseteq \Gamma$.

We will present and illustrate the dynamic proof theory of ALs by means of the ALs $\mathbf{MIO_1^{\cap}}$ and $\mathbf{MIO_1^{\cup}}$, the adaptive counterparts of $\mathrm{out}_1^{\cap}$ and $\mathrm{out}_1^{\cup}$ respectively. In Section 3.2, we move to a more general level and define the adaptive counterparts of all constrained I/O operations.

Let us start with $\mathbf{MIO_1^{\cap}}$. The LLL of this system is the logic $\mathbf{CMIO_1^{+}}$ as defined in Section 2.4 where $\mathbf{1}$ consists of the rules (Z), (WO), (SI) and (AND) which characterize out_1. The set $\Omega_{\bullet}$ of $\mathbf{MIO_1^{\cap}}$-abnormalities is defined as follows:

$$\Omega_{\bullet} =_{\mathrm{df}} \{\bullet A \wedge \neg A \mid A \text{ is of the form } B \rightarrow C\}$$

In what follows, we use $\lightning A$ as an abbreviation for $\bullet A \wedge \neg A$. Abnormalities are generated, for instance, when a conditional is 'triggered' by the input while its consequent violates a constraint. Let, for example $\mathcal{G} = \{(a, \neg b)\}$, $\mathcal{A} = \{a\}$, and $\mathcal{C} = \{b\}$, such that $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} = \{\bullet(a \rightarrow \neg b), \mathsf{in}a, \mathsf{con}b\}$. We show that the abnormality $\lightning(a \rightarrow \neg b)$ is a $\mathbf{CMIO_1}$-consequence of $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}}$.

By (ROC), we know that $\neg\mathsf{out}\neg b$ is derivable from $\mathsf{con}b$. By (MP$_{\Rightarrow}$) and **CL**, it follows that $\neg(\mathsf{in}a \wedge (a \rightarrow \neg b))$. Since $\mathsf{in}a \in \Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}}$, it follows by **CL** that $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO_1}} \bullet(a \rightarrow \neg b) \wedge \neg(a \rightarrow \neg b)$.

The 'motor' that drives the activation of conditionals is the (defeasible) assumption that abnormalities are false. Assume for instance that $\bullet(a \rightarrow b)$. Then since $\vdash_{\mathbf{CMIO_1}} (a \rightarrow b) \vee \neg(a \rightarrow b)$ it follows that $\bullet(a \rightarrow b) \vdash_{\mathbf{CMIO_1}} (a \rightarrow b) \vee \lightning(a \rightarrow b)$. If we can safely assume the second disjunct to be false, then the first must be true. This is—on an intuitive level—how the adaptive proof theory will allow us to activate conditionals. Let us now make this idea formally precise.

Adaptive proofs. A line in an annotated adaptive proof consists of four elements: a line number l, a formula A, a justification (consisting of a list of line numbers and a derivation rule), and a condition Δ. The condition of a line is a (possibly empty) finite set of abnormalities. Intuitively, we interpret a line l at which a formula A is derived on the condition Δ as "At line l of the proof, A is derived on the assumption that all members of Δ are false".

ALs have three generic rules of inference. The first is a premise introduction rule PREM, which allows formulas A from some set of premises Γ to be introduced on the empty condition at any stage in the proof.

PREM If $A \in \Gamma$: $\dfrac{\vdots \quad \vdots}{A \quad \emptyset}$

The second rule is the unconditional rule RU, which allows for the use of all $\mathbf{CMIO_1}$-inferences:

RU If $A_1, \ldots, A_n \vdash_{\mathbf{CMIO_1}} B$:
$$\begin{array}{ll} A_1 & \Delta_1 \\ \vdots & \vdots \\ A_n & \Delta_n \\ \hline B & \Delta_1 \cup \ldots \cup \Delta_n \end{array}$$

The third generic rule of inference is the conditional rule RC. Where Θ is a finite set of abnormalities, let $Dab(\Theta)$ denote the classical disjunction of the members of Θ.[20] The rule RC is defined as follows:

RC If $A_1, \ldots, A_n \vdash_{\mathbf{CMIO_1}} B \vee Dab(\Theta)$
$$\begin{array}{ll} A_1 & \Delta_1 \\ \vdots & \vdots \\ A_n & \Delta_n \\ \hline B & \Delta_1 \cup \ldots \cup \Delta_n \cup \Theta \end{array}$$

RC is the rule that allows for the activation (i.e., the removal of the '$\bullet$'-prefix) and the subsequent detachment of conditionals. For instance, let $\Gamma_2 = \{\bullet(a \to b), \mathsf{in}a\}$. We can start a $\mathbf{MIO}_1^{\cap}$-proof by introducing the members of Γ_2 by means of the rule PREM:

1	$\bullet(a \to b)$	PREM	$\emptyset$
2	$\mathsf{in}a$	PREM	$\emptyset$

By **CL** we know that $\bullet(a \to b) \vdash_{\mathbf{CMIO_1}} (a \to b) \vee \lightning(a \to b)$. Hence we can derive $(a \to b) \vee \lightning(a \to b)$ by an application of RU to line 1:

3	$(a \to b) \vee \lightning(a \to b)$	1; RU	$\emptyset$

Note that the second disjunct of the formula derived at line 3 is a member of $\Omega_\bullet$. By RC, we can move this abnormality to the condition column:

4	$a \to b$	3; RC	$\{\lightning(a \to b)\}$

At stage 5 of the proof, we have derived the formula $a \to b$ on the assumption that the abnormality $\lightning(a \to b)$ is false. The conditional at line 1 has been activated at line 5. By (MP$_\Rightarrow$), we know that $\mathsf{in}a \wedge (a \to b) \vdash_{\mathbf{CMIO_1}} \mathsf{out}b$. Hence we can detach the output $\mathsf{out}b$ by means of RU:

5	$\mathsf{out}b$	2, 4; RU	$\{\lightning(a \to b)\}$

Note that, as required by the definition of RU, the condition of line 5 is carried over to line 4. The example shows how activated conditionals, when triggered by a matching input, can be used to detach the output.

The inference of $\mathsf{out}b$ from $\bullet(a \to b)$ and $\mathsf{in}a$ is *conditional*. It depends on our assumption that $\lightning(a \to b)$ is false. As we will now illustrate, assumptions

[20] "Dab" abbreviates "Disjunction of Abnormalities". If Θ is a singleton $\{A\}$, then $Dab(\Theta) = A$.

made in adaptive proofs are sometimes treated as inadmissible. In such cases, all inferences that depend on the assumptions in question are retracted from the proof.

Retracting inferences. Consider the following adaptive proof from $\Gamma_3 = \{\bullet(a \to b), \mathsf{in}a, \mathsf{con}\neg b\}$:

1	$\bullet(a \to b)$	PREM	$\emptyset$
2	$\mathsf{in}a$	PREM	$\emptyset$
3	$\mathsf{con}\neg b$	PREM	$\emptyset$
4	$\mathsf{out}b$	1,2; RC	$\{\lightning(a \to b)\}$

As there is a constraint prohibiting that b is in the output, it seems that we have jumped to an incorrect conclusion at line 4.[21] In order to deal with such cases, ALs are equipped with a mechanism that determines the retraction or 'marking' of lines of which the condition can no longer be upheld.

Note that, by (ROC), $\mathsf{con}\neg b \vdash_{\mathbf{CMIO_1}} \neg\mathsf{out}b$. By ($\mathrm{MP}_{\Rightarrow}$) and **CL**, $\mathsf{con}\neg b \vdash_{\mathbf{CMIO_1}} \neg(\mathsf{in}a \wedge (a \to b))$. As $\mathsf{in}a, \mathsf{con}\neg b \in \Gamma_3$, it follows that $\Gamma_3 \vdash_{\mathbf{CMIO_1}} \neg(a \to b)$. But then $\Gamma_3 \vdash_{\mathbf{CMIO_1}} \lightning(a \to b)$, which falsifies our assumption made at line 4:

4	$\mathsf{out}b$	1,2; RC	$\{\lightning(a \to b)\}\checkmark^5$
5	$\lightning(a \to b)$	1–3; RU	$\emptyset$

At line 4 we mistakenly assumed that $\lightning(a \to b)$ is false. Once our assumption is shown to be inadmissible (at line 5), line 4 is marked (using the symbol $\checkmark^5$), which means that this inference is now withdrawn from the proof.

The retraction mechanism is governed by a *marking definition*, which depends on the adaptive strategy. In the remainder of this section, we present two such strategies and their respective marking definitions. They correspond to various 'styles' of reasoning —skeptical versus credulous—, as we will now illustrate.

Consider a proof from $\Gamma_4 = \{\bullet(a \to c), \bullet(b \to d), \mathsf{in}a, \mathsf{in}b, \mathsf{con}\neg(c \wedge d)\}$:

1	$\bullet(a \to c)$	PREM	$\emptyset$
2	$\bullet(b \to d)$	PREM	$\emptyset$
3	$\mathsf{in}a$	PREM	$\emptyset$
4	$\mathsf{in}b$	PREM	$\emptyset$
5	$\mathsf{con}\neg(c \wedge d)$	PREM	$\emptyset$
6	$\mathsf{out}c$	1,3; RC	$\{\lightning(a \to c)\}$

[21] The application of RC at line 4 is valid since $\bullet(a \to b), \mathsf{in}a \vdash_{\mathbf{CMIO_1}} \mathsf{out}b \vee \lightning(a \to b)$.

7	$\mathsf{out}d$	2,4; RC	$\{\lightning(b \to d)\}$

As out is a normal modal operator, we can aggregate the formulas derived at lines 6 and 7:

8	$\mathsf{out}(c \wedge d)$	6,7; RU	$\{\lightning(a \to c), \lightning(b \to d)\}$

Clearly, something went wrong here. As there is a constraint prohibiting that the conjunction of c and d is in the output, we were too hasty in deriving $\mathsf{out}(c \wedge d)$. And indeed, although no abnormality is by itself derivable from the premises, the disjunction of abnormalities $\lightning(a \to c) \vee \lightning(b \to d)$ is a $\mathbf{CMIO_1}$-consequence of Γ_4:[22]

9	$\lightning(a \to c) \vee \lightning(b \to d)$	1–5; RC	$\emptyset$

At stage 9 of the proof, we know that one of the abnormalities $\lightning(a \to c)$ or $\lightning(b \to d)$ holds, but we lack the information to determine which one. Clearly, line 8 should be retracted, as at that line we assumed *both* abnormalities to be false. But what about lines 6 and 7? If only the abnormality $\lightning(a \to c)$ turns out to be derivable, then our assumption at line 7 that $\lightning(b \to d)$ is false can safely be upheld. If only $\lightning(b \to d)$ turns out to be derivable, then the same holds for our assumption at line 6 that $\lightning(a \to c)$ is false.

In view of this information a *skeptical* reasoner may consider *both* assumptions too strong. As a result, she would retract lines 6 and 7 from the proof. A more *credulous* reasoner, on the other hand, may continue to reason on one assumption or the other, as none of them has been falsified beyond doubt. The credulous reasoner, then, would leave lines 6 and 7 unmarked.

The minimal abnormality strategy explicates the more skeptical reasoning, while the normal selections strategy explicates the more credulous reasoning. $\mathbf{MIO_1^{\cap}}$ uses the minimal abnormality strategy..

The minimal abnormality strategy. The marking definition for the minimal abnormality strategy requires some further terminology. A *Dab*-formula $Dab(\Delta)$ derived at stage s is *minimal* at that stage iff $Dab(\Delta)$ is derived on the empty condition, and no $Dab(\Delta')$ with $\Delta' \subset \Delta$ is derived on the empty condition at stage s. Note that in the above proof, the only (minimal) *Dab*-formula at stage 9 is the disjunction derived on line 9.

---

[22] By (ROC), (†) $\mathsf{con}\neg(c \wedge d) \vdash_{\mathbf{CMIO_1}} \neg\mathsf{out}(c \wedge d)$. By (MP$_\Rightarrow$) and **CL**, $\mathsf{in}a \vdash_{\mathbf{CMIO_1}} \neg(a \to c) \vee \mathsf{out}c$ and $\mathsf{in}b \vdash_{\mathbf{CMIO_1}} \neg(b \to d) \vee \mathsf{out}d$. Altogether, $\mathsf{in}a, \mathsf{in}b \vdash_{\mathbf{CMIO_1}} \neg(a \to c) \vee \neg(b \to d) \vee (\mathsf{out}c \wedge \mathsf{out}d)$. By **KD**, $\mathsf{in}a, \mathsf{in}b \vdash_{\mathbf{CMIO_1}} \neg(a \to c) \vee \neg(b \to d) \vee \mathsf{out}(c \wedge d)$. But then, by (†) and **CL**, $\mathsf{con}\neg(c \wedge d), \mathsf{in}a, \mathsf{in}b \vdash_{\mathbf{CMIO_1}} \neg(a \to c) \vee \neg(b \to d)$. By **CL** again, $\Gamma_4 \vdash_{\mathbf{CMIO_1}} (\bullet(a \to c) \wedge \neg(a \to c)) \vee (\bullet(b \to d) \wedge \neg(b \to d))$.

Let a *choice set* of $\Sigma = \{\Delta_1, \Delta_2, \ldots\}$ be a set that contains one element out of each member of Σ. A *minimal choice set* of Σ is a choice set of Σ of which no proper subset is a choice set of Σ. Where $Dab(\Delta_1), Dab(\Delta_2), \ldots$ are the minimal Dab-formulas at stage s of a proof and $\Sigma_s(\Gamma) = \{\Delta_1, \Delta_2, \ldots\}$, $\Phi_s(\Gamma)$ is the set of minimal choice sets of $\Sigma_s(\Gamma)$. So in the above proof, we have $\Phi_9(\Gamma_4) = \{\{\lightning(a \to c)\}, \{\lightning(b \to d)\}\}$.

In order to understand the marking for minimal abnormality, recall first: if A is derived on the condition Δ this encodes the assumption that none of the abnormalities in Δ is true. The minimal choice sets Θ in $\Phi_s(\Gamma)$ represent minimally abnormal interpretations of all the disjunctions of abnormalities that have been derived on the empty condition so far. A formula A is considered successfully derived at stage s in case for each $\Theta \in \Phi_s(\Gamma)$ it is derived on an assumption (expressed by a condition Δ_Θ) that is not violated in Θ, i.e., $\Theta \cap \Delta_\Theta = \emptyset$. This is expressed in requirement (ii) of the following definition. Requirement (i) makes sure that lines get marked at which A is derived on an assumption Δ that is violated in all minimal abnormal choice sets.

Definition 10. Where A is derived at line l of a proof from Γ on a condition Δ, line l is marked at stage s iff

(i) there is no $\Theta \in \Phi_s(\Gamma)$ such that $\Theta \cap \Delta = \emptyset$, or

(ii) for some $\Theta \in \Phi_s(\Gamma)$, there is no line at which A is derived on a condition Δ' for which $\Theta \cap \Delta' = \emptyset$.

In view of this definition it follows that all of lines 6–8 are marked. For line 8, this is obvious: clause (i) of Definition 10 clearly obtains. For lines 6 and 7, clause (i) fails, but clause (ii) holds. For instance, there is no line in the proof at which $\mathsf{out}c$ is derived on a condition that has an empty intersection with $\{\lightning(a \to c)\}$.

We now illustrate why Definition 10 refers to other lines than l in its clause (ii). Suppose we continue our proof, weakening the output obtained at lines 7 and 8:

10	$\mathsf{out}(c \vee d)$	6; RU	$\{\lightning(a \to c)\}$
11	$\mathsf{out}(c \vee d)$	7; RU	$\{\lightning(b \to d)\}$

Note that, since we have not derived any new Dab-formulas, $\Phi_{11}(\Gamma_4) = \Phi_{11}(\Gamma_4) = \{\{\lightning(a \to c)\}, \{\lightning(b \to d)\}\}$. Note that for each $\Theta \in \Phi_{11}(\Gamma_4)$, $\mathsf{out}(c \vee d)$ is derived on a condition that does not intersect with it. Hence, by Definition 10, lines 10 and 11 are unmarked at stage 11 of the proof.

Two notions of derivability. In adaptive proofs, markings may come and go. Certain lines may be marked at some stage s, while they may not be marked at a later stage s'. This can be so for several reasons:

(i) New minimal *Dab*-formulas may be derived at stage s'—hence certain assumptions which were tenable at s are no longer tenable at s'.

(ii) Some $Dab(\Delta)$ may be a minimal *Dab*-formula at stage s, but not at stage s'—hence certain assumptions which were not tenable at stage s may again become tenable at stage s' and vice versa.

(iii) Where A was already derived on marked lines $l_1, \ldots, l_n$ on the respective conditions $\Delta_1, \ldots, \Delta_n$ at stage s, it may be derived on additional conditions $\Theta_1, \ldots, \Theta_m$ at stage s' which may lead to the unmarking of some of the lines $l_1, \ldots, l_n$.

For an example of (iii), consider again the last proof from this section. At stage 10, $\mathsf{out}(c \vee d)$ is only derived on the condition $\{\lightning(a \to c)\}$, which intersects with the minimal choice set $\{\lightning(a \to c)\}$. Hence, at this stage, line 10 is marked. However, at the next stage, we have derived $\mathsf{out}(c \vee d)$ also on the condition $\{\lightning(b \to d)\}$. As a result, line 10 is unmarked at stage 11.

A dynamic, stage-dependent notion of derivability is easy to define:

Definition 11. A formula A has been *derived at stage s* of an adaptive proof iff, at that stage, A is the second element of some unmarked line l.

Apart from the stage-dependent derivability relation, we also need a stage-independent, 'final' notion of derivability in order to define a syntactic consequence relation for our logics.

Definition 12. A is *finally derived* from Γ at line l of a proof at a finite stage s iff (i) A is the second element of line l, (ii) line l is not marked at stage s, and (iii) every extension of the proof in which line l is marked can be further extended in such a way that line l is unmarked.[23]

[23]Definition 12 has a game-theoretic flavor to it. In [7], this definition is interpreted as a two-player game in which the proponent has a winning strategy in case she has a reply to every counterargument by her opponent. As is clear from the definition, final derivability is not established within the dynamic proof itself, but rather by meta-reasoning (one has to quantify over possible extensions of the proof). For some this may be unsatisfactory (e.g., if one is interested in automated proving). We refer to [5] where a technique is presented that integrates a decision procedure for final derivability into dynamic proofs for the adaptive logic $\mathbf{CLuN}^{\mathbf{r}}$ by means of turning proofs goal directed [8]. Adjusting this technique for the systems presented in this paper is left for future work.

Definition 13. $\Gamma \vdash_{\mathbf{MIO}_1^{\cap}} A$ (A is finally $\mathbf{MIO}_1^{\cap}$-derivable from Γ) iff A is finally derived at a line of an $\mathbf{MIO}_1^{\cap}$-proof from Γ.

The disjunction $\lightning(a \to c) \vee \lightning(b \to d)$ is the only *minimal Dab-consequence* of Γ_4, i.e. the only minimal *Dab*-formula that is $\mathbf{CMIO_1}$-derivable from Γ_4. Consequently, the set $\Phi_{11}(\Gamma_4) = \{\{\lightning(a \to c)\}, \{\lightning(b \to d)\}\}$ will remain stable in any extension of the proof, such that lines 10 and 11 remain unmarked whatever happens next. By Definition 12, $\mathsf{out}(c \vee d)$ is finally derived from Γ_4. By Definition 13, $\Gamma_4 \vdash_{\mathbf{MIO}_1^{\cap}} \mathsf{out}(c \vee d)$. It is safely left to the reader to check that $\Gamma_4 \nvdash_{\mathbf{MIO}_1^{\cap}} \mathsf{out}c, \Gamma_4 \nvdash_{\mathbf{MIO}_1^{\cap}} \mathsf{out}d$, and $\Gamma_4 \nvdash_{\mathbf{MIO}_1^{\cap}} \mathsf{out}(c \wedge d)$.

The normal selections strategy. We saw how the minimal abnormality strategy is rather skeptical in its treatment of constrained outputs. In our example we were able to derive the minimal *Dab*-consequence $\lightning(a \to c) \vee \lightning(b \to d)$. This means that both conditionals together are not $\mathbf{CMIO_1}$-consistent with our given inputs and constraints. Since the disjunction is minimal, both $a \to c$ and $b \to d$ taken individually are consistent with the inputs and constraints. Nevertheless, a line whose formula is only derived on the condition $\{\lightning(a \to c)\}$ or only on the condition $\{\lightning(b \to d)\}$ is marked according to minimal abnormality (e.g., lines 6 and 7). This motivates another approach according to which an assumption is considered admissible in case its associated set of conditionals is consistent with the given inputs and constraints. That is to say: where $\Delta = \{\lightning(A_1 \to B_1), \ldots, \lightning(A_n \to B_n)\}$ is the condition of the line l, this line is marked only if $Dab(\Delta)$ is derived on the empty condition since this expresses that the set of conditionals $\{(A_1 \to B_1), \ldots, (A_n \to B_n)\}$ is not consistent with the given input and constraints.

Let us spell out this more credulous alternative by means of the lower limit logic $\mathbf{CMIO_1}$. We now use the so-called *normal selections* strategy. The AL that has $\mathbf{CMIO_1}$ as its LLL, $\Omega_\bullet$ (as defined above) as its set of abnormalities, and normal selections as its strategy is called $\mathbf{MIO}_1^{\cup}$. Thus, $\mathbf{MIO}_1^{\cup}$ differs from $\mathbf{MIO}_1^{\cap}$ only in its use of a different strategy and, hence, a different marking definition.

The marking definition for the normal selections strategy is straightforward:

Definition 14. Line l is marked at stage s iff, where Δ is the condition of line l, $Dab(\Delta)$ has been derived on the empty condition at stage s.

We consider again the proof from Γ_4, but this time we mark lines according to the normal selections strategy.

1	$\bullet(a \to c)$	PREM	$\emptyset$
2	$\bullet(b \to d)$	PREM	$\emptyset$
3	$\mathsf{in}a$	PREM	$\emptyset$
4	$\mathsf{in}b$	PREM	$\emptyset$
5	$\mathsf{con}\neg(c \wedge d)$	PREM	$\emptyset$
6	$\mathsf{out}c$	1,3; RC	$\{\lightning(a \to c)\}$
7	$\mathsf{out}d$	2,4; RC	$\{\lightning(b \to d)\}$
8	$\mathsf{out}(c \wedge d)$	6,7; RU	$\{\lightning(a \to c), \lightning(b \to d)\}\checkmark^9$
9	$\lightning(a \to c) \vee \lightning(b \to d)$	1–5; RC	$\emptyset$
10	$\mathsf{out}(c \vee d)$	6; RU	$\{\lightning(a \to c)\}$
11	$\mathsf{out}(c \vee d)$	7; RU	$\{\lightning(b \to d)\}$

Lines 6, 7, 10 and 11 remain unmarked in view of Definition 14, as their conditions have not been derived in the proof. Line 8 is marked since $Dab(\{\lightning(a \to c), \lightning(b \to d)\})$ is derived at line 9.

The stage-dependent and stage-independent criteria for derivability (Definitions 11, 12) are equally applicable to $\mathbf{MIO_1^{\cup}}$.[24] A syntactic consequence relation for $\mathbf{MIO_1^{\cup}}$ is readily defined by adjusting Definition 13 as follows:

Definition 15. $\Gamma \vdash_{\mathbf{MIO_1^{\cup}}} A$ (A is finally $\mathbf{MIO_1^{\cup}}$-derivable from Γ) iff A is finally derived at a line of an $\mathbf{MIO_1^{\cup}}$-proof from Γ.

Since $\Gamma_4 \nvdash_{\mathbf{CMIO_1}} \lightning(a \to c)$ and $\Gamma_4 \nvdash_{\mathbf{CMIO_1}} \lightning(b \to d)$, there is no possible extension of our proof in which any of lines 6, 7, 10 and 11 are marked. By Definition 15, $\Gamma_4 \vdash_{\mathbf{MIO_1^{\cup}}} \mathsf{out}c$, $\Gamma_4 \vdash_{\mathbf{MIO_1^{\cup}}} \mathsf{out}d$, $\Gamma_4 \vdash_{\mathbf{MIO_1^{\cup}}} \mathsf{out}(c \vee d)$, and $\Gamma_4 \nvdash_{\mathbf{MIO_1^{\cup}}} \mathsf{out}(c \wedge d)$.

Some more words on proof dynamics. I/O functions equip us with consequence relations. In our characterization, the consequences of a given I/O function correspond to the finally derivable formulas (Definition 12). Dynamic proofs come with an additional notion of derivability, namely derivability at a stage (Definition 11).

This corresponds to the fact that non-monotonic resp. defeasible reasoning has two flavors and for many applications it seems natural to model both in an integrated way. From an abstract point of view non-monotonicity means that the output of a function is disproportionate to its input: more input may result in less output. The first and most commonly discussed

[24]In fact, the definition for final derivability can be simplified since a line that is once marked will not be unmarked at any later stage of the proof according to the normal selections strategy. Hence, we can simply define: A is finally derived on some unmarked line l at stage s if the proof cannot be extended in such a way that line l is marked.

flavor concerns the consequence function and the fact that adding premises may lead to a loss of conclusions. In adaptive proofs this is the dynamics behind final derivability (Definition 12). The second flavor of non-monotonicity we have with derivability-at-a-stage (Definition 11). Derivability-at-a-stage viewed as a function takes as its input a (possibly incomplete) analysis of a given premise set (represented by the proof at the given stage) and outputs formulas that are considered derived. Given a more refined analysis of the premise set at a later stage, some formulas may cease to be derived (formulas at unmarked lines). Let us apply this insight to I/O logics.

Consider a case where some I/O consequence B of $\langle \mathcal{G}, \mathcal{A}, \mathcal{C} \rangle$ ceases to be an I/O consequence of $\langle \mathcal{G}', \mathcal{A}', \mathcal{C}' \rangle$ (where $\mathcal{G} \subseteq \mathcal{G}'$, $\mathcal{A} \subseteq \mathcal{A}'$ and $\mathcal{C} \subseteq \mathcal{C}'$). An adequate proof theory is expected to reflect this dynamics: $\mathsf{out}B$ will be (finally) derivable from $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}}$, while it will not be (finally) derivable in a proof from $\Gamma^{\mathcal{G}',\mathcal{A}',\mathcal{C}'}$. In the presented proof format this dynamics that is relative to adding premises is 'internalized' so that it is also relative to the growing analysis of the premises which is represented by the proof at a given stage. It works as follows: At each stage the given premises will be analysed to some degree, depending on which premises have already been introduced and depending on what rules have been already applied to them. Derivability-at-a-stage equips us with a retraction mechanism that is a function of this degree of analysis of the given premise set. This way the very rationale behind the non-monotonicity of I/O logics, namely that in face of new conflicts previous outputs may cease to be outputs, becomes a structural property of the presented dynamic proofs (and not just of the consequence relation).

This property of the presented proof theory is important since it models the dynamics of a reasoner in one of the following two situations.

First, we have a reasoner who is confronted with new factual or normative information in view of which previously drawn inferences have to be retracted: clearly a situation in which real life reasoners often find themselves. Pollock dubs this *diachronic defeasibility* [37] and Batens speaks of the *external dynamics* of defeasible reasoning [6].

Another situation is that in which the given information is very complex and a reasoner has practical constraints (e.g., concerning time) which make it interesting for her to 'jump' to defeasible conclusions based on a given analysis of the premises (which, given the practical constraints, she considers to be sufficient). Pollock dubs this *synchronic defeasibility* and Batens speaks of the *internal dynamics* of reasoning.

In both situations it holds that, instead of starting the reasoning process from scratch whenever new information enters the picture, a real life reasoner will build on some of her previous inferences, use them to draw new inferences, and informed by these, will retract certain previous inferences. Retraction is thus *local and case-specific*.[25] The dynamic proof theory presented in this paper is adequate for such application contexts in that a user need not analyze the premises exhaustively before she can come to a (defeasible) conclusion and her conclusions can be formed in a computationally tractable way.

In situations in which a reasoner is confronted with (a) a finite set of (relevant) facts and (b) a finite set of conditional norms, and (c) in which there are no time-related or other types of constraints that motivate a reasoner to proceed in a defeasible manner, a rational reasoner will make inferences in a more cautious and controlled way. She will first carefully identify all normative conflicts (i.e., all minimal *Dab*-consequences), and only then and informed by the former, derive the output in such a way that no retraction of previously derived formulas is necessary. Here it is important to notice that nothing prevents a reasoner from applying this or some other heuristics to the dynamic proofs characterized above. Indeed, she may first introduce all the facts in $\mathcal{A}$, all constraints in $\mathcal{C}$ and all conditional norms in $\mathcal{G}$ and on this basis derive the minimal *Dab*-consequences. The latter are all $Dab(\Delta)$ for which $\Delta \in \Sigma(\Gamma)$ where $\Sigma(\Gamma) = \{\Delta \subseteq \Omega_{\bullet} \mid \Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} Dab(\Delta)$ while $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \nvdash_{\mathbf{CMIO}_{\mathsf{R}}} Dab(\Delta')$ for all $\Delta' \subset \Delta\}$. This way she brings the proof to a stage s in which $\Sigma_s(\Gamma) = \Sigma(\Gamma)$.[26] Clearly, any (extension of a) proof in such a stage is not dynamic in view of the following fact[27]:

Fact 4. *If $\Sigma(\Gamma) = \Sigma_s(\Gamma)$ and A is derived at the finite stage s, then A is finally derived.*

Irrespective of whether we are in a situation in which (a)–(c) hold, it is interesting to have a criterion to decide whether a formula that is derived at a stage is finally derived. The previous fact equips us with such a criterion.

[25] See also [51] where a similar case is made for belief revision.

[26] Note that $\mathbf{CMIO}_{\mathsf{R}}$ is decidable. Hence, for each $\mathcal{G}' \subseteq \mathcal{G}$ we can check whether $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \bigvee_{(A,B)\in\mathcal{G}'} \lightning(A \to B)$ and whether for all $(C,D) \in \mathcal{G}'$, $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \nvdash_{\mathbf{CMIO}_{\mathsf{R}}} \bigvee_{(A,B)\in\mathcal{G}'\setminus\{(C,D)\}} \lightning(A \to B)$. Thus, there are no computational obstacles in bringing the proof to this stage. One may also make use of goal-directed decision procedures for adaptive logics as presented in e.g. [8].

[27] Fact 4 holds for any AL defined as in Section 3.1 (i.e., for any AL in the standard format).

3.2. Adaptive Characterizations of Constrained I/O Logics

Generic definition. We now move to a more abstract level, defining the adaptive logics that characterize the constrained I/O functions from Section 2.3.

Definition 16. Where $\dagger \in \{\cap, \cup\}$ and $\ddagger \in \{+, -\}$, $\mathbf{MIO}^{\ddagger,\dagger}_{\mathsf{R}}$ is the AL defined from

1. The lower limit logic $\mathbf{CMIO}^{\ddagger}_{\mathsf{R}}$ (see Definition 8)
2. The set of abnormalities $\Omega_{\bullet} = \{\lightning(A \to B) \mid A, B \in \mathcal{W}\}$
3. The strategy:
   - *minimal abnormality* if $\dagger = \cap$
   - *normal selections* if $\dagger = \cup$

We write $\mathbf{MIO}^{\dagger}_{\mathsf{R}}$ whenever we refer to any of the two variants $\mathbf{MIO}^{-,\dagger}_{\mathsf{R}}$ or $\mathbf{MIO}^{+,\dagger}_{\mathsf{R}}$.

By adjusting the subscripts in the definitions of the generic inference rules from the previous section, we readily obtain a proof theory for $\mathbf{MIO}^{\dagger}_{\mathsf{R}}$. By using the appropriate marking definition (see Definitions 10 and 14) and applying Definitions 12 and 13, we obtain a consequence relation.

The following is proven in Appendix 3:

Theorem 3. *Where* $\dagger \in \{\cap, \cup\}$, *and* $\mathcal{A}$ *is* $\mathbf{CL}$*-consistent or* (F) $\in \hat{\mathsf{R}}$*:*

1. $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{MIO}^{-,\dagger}_{\mathsf{R}}} \mathsf{out}A$ *iff* $A \in \mathrm{out}^{\dagger}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})$
2. *Where (WO), (Z)* $\in \hat{\mathsf{R}}$, $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{MIO}^{+,\dagger}_{\mathsf{R}}} \mathsf{out}A$ *iff* $A \in \mathrm{out}^{\dagger}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})$

Like Theorem 2, Theorem 3 does not cover the border case where $\mathcal{A}$ is inconsistent. As explained before, such cases are trivialized by our logics whereas (in case (ID) $\notin \hat{\mathsf{R}}$) neither $\mathrm{out}^{\cap}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})$ nor $\mathrm{out}^{\cup}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})$ need be trivial.[28]

Proof outline. In the remainder of this section, we broadly explain how we prove the equivalence set out in Theorem 3 between the adaptive consequence relations on the one hand, and the constrained I/O operations on the other. The reader is kindly referred to Appendix 3 for more details.

Let us first deal with the special case in which there are no maxfamilies.

Lemma 1. *Where* $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C}$ *is* $\mathbf{CL}$*-inconsistent:* $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \bot$ *and hence* $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{MIO}^{\cap}_{\mathsf{R}}} \bot$ *and* $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{MIO}^{\cup}_{\mathsf{R}}} \bot$.

[28] Take the simple example where $\mathcal{G} = \mathcal{C} = \emptyset$ and $\mathcal{A} = \{\bot\}$ in which case $\mathrm{out}^{\cap}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C}) = \mathrm{out}^{\cup}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C}) = \mathrm{Cn}_{\mathbf{CL}}(\emptyset)$.

PROOF. Suppose $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C} \vdash_{\mathbf{CL}} \bot$. If $\mathcal{A} \vdash_{\mathbf{CL}} \bot$ then $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \bot$ since in is a $\mathbf{KD}$-modality. Suppose $\mathcal{A} \nvdash_{\mathbf{CL}} \bot$. By Remark 1, there is an A for which $\mathcal{C} \vdash_{\mathbf{CL}} A$ and $\neg A \in \mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A})$. By Theorem 2, $\Gamma^{\emptyset,\mathcal{A}} \vdash_{\mathbf{MIO}_{\mathsf{R}}} \mathsf{out}\neg A$. Thus by monotonicity and since $\mathbf{CMIO}_{\mathsf{R}}$ extends $\mathbf{MIO}_{\mathsf{R}}$, $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \mathsf{out}\neg A$. Also, since con is a $\mathbf{KD}$-modality, $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \mathsf{con}A$ and thus by (ROC), $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \neg\mathsf{out}\neg A$. Altogether, $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \bot$. ■

As a consequence, we know now that Theorem 3 holds whenever $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C}$ is $\mathbf{CL}$-inconsistent. For the other case we need a more involved argument for which it is useful to introduce some additional notation.

Notation. Let $\Theta \subseteq_f \Delta$ denote that Θ is a finite subset of Δ. Where Δ is a set of formulas of the form $A \to B$, let $\Omega_\bullet(\Delta) = \{\lightning(A \to B) \mid A \to B \in \Delta\}$. Where $\mathcal{G}$ is a set of I/O pairs we will just write $\Omega_\bullet(\mathcal{G})$ instead of $\Omega_\bullet(\{A \to B \mid (A, B) \in \mathcal{G}\})$.

Recall from Section 3.1 that $Dab(\Delta)$ is a *Dab-consequence of* Γ iff $\Gamma \vdash_{\mathbf{CMIO}_{\mathsf{R}}} Dab(\Delta)$. It is a *minimal Dab-consequence of* Γ iff there is no $\Delta' \subset \Delta$ such that $\Gamma \vdash_{\mathbf{CMIO}_{\mathsf{R}}} Dab(\Delta')$. We let $\Sigma(\Gamma)$ be the set of all Δ for which $Dab(\Delta)$ is a minimal Dab-consequence of Γ and $\Phi(\Gamma)$ is the set of all minimal choice sets of $\Sigma(\Gamma)$.

Let us start with the operations $\mathrm{out}_{\mathsf{R}}^{\cap}$ and logics $\mathbf{MIO}_{\mathsf{R}}^{\cap}$. The first theorem which we need has been established for all ALs with the minimal abnormality strategy in [6]:

THEOREM 4. ([6], Theorem 8) $\Gamma \vdash_{\mathbf{MIO}_{\mathsf{R}}^{\cap}} A$ *iff for all* $\Theta \in \Phi(\Gamma)$, *there is a* $\Delta \subseteq_f \Omega_\bullet - \Theta$ *such that* $\Gamma \vdash_{\mathbf{CMIO}_{\mathsf{R}}} A \vee Dab(\Delta)$.[29]

For the specific case where $\Gamma = \Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}}$ for some $\mathcal{G}, \mathcal{A}, \mathcal{C}$, we can further strengthen Theorem 4 to Corollary 1 in view of the following lemma:

LEMMA 2. *If* $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} A \vee Dab(\Delta)$, *then* $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} A \vee Dab(\Delta \cap \Omega_\bullet(\mathcal{G}))$.

COROLLARY 1. $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{MIO}_{\mathsf{R}}^{\cap}} A$ *iff for all* $\Theta \in \Phi(\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}})$, *there is a* $\Delta \subseteq_f \Omega_\bullet(\mathcal{G}) - \Theta$ *such that* $\Gamma \vdash_{\mathbf{CMIO}_{\mathsf{R}}} A \vee Dab(\Delta)$.

The second property which we need is the following:

THEOREM 5. *Where* $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C}$ *is* $\mathbf{CL}$*-consistent and* $\mathcal{A}$ *is* $\mathbf{CL}$*-consistent or* (F) $\in \hat{\mathsf{R}}$*:*

$$\Phi(\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}}) = \{\Omega_\bullet(\mathcal{G} - \mathcal{H}) \mid \mathcal{H} \in \mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})\}$$

[29] In this theorem and the results below we adopt the following writing convention: where $\Delta = \emptyset$, "$\vee Dab(\Delta)$" denotes the empty string.

The proof of this theorem is intricate—see Proof of Theorem 5 in Appendix 3. For instance, consider $\mathcal{A} = \{a, b\}, \mathcal{G} = \{(a, c), (b, d), (a \wedge b, e)\}, \mathcal{C} = \{\neg(c \wedge d)\}$ from Example 1 and the corresponding premise set $\Gamma_5 = \{\mathsf{in}a, \mathsf{in}b, \bullet(a \to c), \bullet(b \to d), \bullet((a \wedge b) \to e), \mathsf{con}\neg(c \wedge d)\}$. As we saw before, $\mathsf{maxfamily}(\mathcal{G}, \mathcal{A}, \mathcal{C}) = \{\{(a, c), (a \wedge b, e)\}, \{(b, d), (a \wedge b, e)\}\}$. On the other hand, $\Phi(\Gamma_5) = \{\{\lightning(b \to d)\}, \{\lightning(a \to c)\}\}$. We leave it to the reader to check that this conforms to Theorem 5.

The third ingredient for the meta-proof is the following theorem, which links $\mathrm{out}_{\mathsf{R}}$ to $\mathbf{CMIO}_{\mathsf{R}}$:

THEOREM 6. *Where* $\mathcal{H} \subseteq \mathcal{G}$*,* $\mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A}) \cup \mathcal{C}$ *is* **CL***-consistent, and* $\mathcal{A}$ *is* **CL***-consistent or* (F) $\in \hat{\mathsf{R}}$*:* $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A})$ *iff there is a* $\Delta \subseteq_f \Omega_\bullet(\mathcal{H})$ *such that* $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \mathsf{out}A \vee Dab(\Delta)$*.*

With these properties at our disposal, the proof of Theorem 3 for the case $\dagger = \cap$ becomes relatively short. Suppose $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C}$ **CL**-consistent and $\mathcal{A}$ is **CL**-consistent or (F) $\in \hat{\mathsf{R}}$. The following five properties are equivalent by Corollary 1 (items 1 and 2), Theorem 5 (items 2 and 3), Theorem 6 (item 3 and 4) and Definition 6 (item 4 and 5):

1. $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{MIO}^{\cap}_{\mathsf{R}}} \mathsf{out}A$;
2. for all $\Theta \in \Phi(\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}})$, there is a $\Delta \subseteq_f \Omega_\bullet(\mathcal{G}) - \Theta$ such that $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \mathsf{out}A \vee Dab(\Delta)$;
3. for all $\mathcal{H} \in \mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})$, there is a $\Delta \subseteq_f \Omega_\bullet(\mathcal{H})$ for which $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_{\mathsf{R}}} \mathsf{out}A \vee Dab(\Delta)$[30];
4. for all $\mathcal{H} \in \mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})$, $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A})$;
5. $A \in \mathrm{out}^{\cap}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})$.

For the full join constrained output functions, $\mathrm{out}^{\cup}_{\mathsf{R}}$, we can apply basically the same reasoning, relying on the following variant of Theorem 4:

THEOREM 7. ([45], Theorem 2.8.3) $\Gamma \vdash_{\mathbf{MIO}^{\cup}_{\mathsf{R}}} A$ *iff for some* $\Theta \in \Phi(\Gamma)$*, there is a* $\Delta \subseteq_f \Omega_\bullet - \Theta$ *such that* $\Gamma \vdash_{\mathbf{CMIO}_{\mathsf{R}}} A \vee Dab(\Delta)$*.*

In view of this fact, it suffices to replace "for all" by "for some" and the reference to Definition 6 with Definition 7 in the above proof, in order to obtain a proof for the case where $\dagger = \cup$.

[30] In the special case in which $\mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C}) = \{\emptyset\}$ our Δ will be empty. Recall that then "$\vee Dab(\Delta)$" denotes the empty string.

4. Expressive Power: Going Beyond I/O Logic

By considering complex formulas other than $\mathsf{in}\, A$ and $\mathsf{out}\, B$ on the left resp. the right side of $\Rightarrow$, our framework enables us to express more facets of normative reasoning than I/O logic. In this section we explore some of the possibilities.

We first note that in the I/O formalism we have only two ways to introduce negations into I/O pairs (A, B):

(i) negating the input: $(\neg A, B)$ which may be read e.g. as "The input $\neg A$ is a reason to have B in the output."

(ii) negating the output: $(A, \neg B)$ which may be read e.g. as "The input A is a reason to have $\neg B$ in the output."

In our framework these cases are covered by $\mathsf{in}\neg A \Rightarrow \mathrm{out} B$ and $\mathsf{in} A \Rightarrow \mathrm{out}\neg B$ respectively. Linguists and logicians have pointed out that allowing for other uses of negation in the context of conditionals significantly enriches the expressiveness and makes the formal model more apt to capture argumentation and defeasible reasoning [1,2]. These authors mention, for instance,[31]

(iii) "The input A is *not* a reason to have B in the output."

(iv) "The input A is a reason *against* having B in the output."

In our framework we can cover these cases by

(iii') $\neg(\mathsf{in}\, A \Rightarrow \mathsf{out}\, B)$ and

(iv') $\mathsf{in}\, A \Rightarrow \neg\mathsf{out}\, B$ respectively.

Moreover, we can express, e.g.,

(v) $\neg\mathsf{in}\, A \Rightarrow \mathsf{out}\, B$: "*Not* having A in the input is a reason to have B in the output."

More generally, we may allow for other complex formulas on the left side of $\Rightarrow$. We now demonstrate how the additional expressive power is useful for the modelling of various central notions in deontic logic such as sanctions, violations, exceptions, and permissions.

[31] The presentation is adjusted to our formalism.

Violations and sanctions. It has been noted in [19] that in I/O logics it is not possible to express violations and sanctions.[32] The authors use the following example:

1. In general you ought not to have a dog $(\top, \neg d)$, and
2. If you have a dog and the obligation to not have a dog is actual, then you ought to pay a fine.

(2) cannot be expressed by means of a normal I/O pair since the condition includes deontic information. In our setting we can express (2) with ease. For instance, where f represents the paying of a fine, we can express (2) by $(\mathsf{out}\neg d \wedge \mathsf{in}\, d) \Rightarrow \mathsf{out}\, f$. Altogether, the example is then phrased as $\Gamma_{\text{vio}} = \{\mathsf{in}\, d, \bullet(\mathsf{in}\, \top \Rightarrow \mathsf{out}\neg d), \bullet((\mathsf{out}\neg d \wedge \mathsf{in}\, d) \Rightarrow \mathsf{out}\, f)\}$. Since now we do not only want to 'actualize' formulas of the form $\bullet(\mathsf{in}\, A \Rightarrow \mathsf{out}\, B)$ but more generally formulas of the form $\bullet(A \Rightarrow B)$, we generalize $\Omega_\bullet$ to $\Omega_\bullet^+ = \{\bullet A \wedge \neg A \mid A \in \mathcal{W}^c\}$.

With this generalization we get, for instance, $\Gamma_{\text{vio}} \vdash_{\mathbf{CMIO}^{\cap}_{\mathsf{R}}} \mathsf{out}\neg d \wedge \mathsf{out} f$ where R is the rule set of out_1 consisting of (WO), (AND) and (SI).

Dealing with exceptions. We can also express combinations of positive and negative conditions such as

> $(\neg\mathsf{in}\, A \wedge \mathsf{in}\, A') \Rightarrow \mathsf{out}\, B$: "*Not* having A in the input *and* having A' in the input is a reason to have B in the output."

This can be used to explicitely express exceptions as e.g.,

> $(\neg\mathsf{in}\, a \wedge \mathsf{in}\, m) \Rightarrow \mathsf{out}\, \neg f$ which may express that being served a meal (m) which is not asparagus (a) we're obliged not to eat with fingers f.

In non-monotonic logics the modelling of conditionals that allow for exceptions often takes place under a closed world assumption: while positive (relevant) information is presented (like the fact that a meal is served $\mathsf{in}\, m$), negative information is omitted (like the fact that it is not the case that asparagus is served $\neg\mathsf{in}\, a$). For instance, given $\Gamma_{\text{defneg}} = \{\mathsf{in}\, m\} \cup \{\bullet((\mathsf{in}\, m \wedge \neg\mathsf{in}\, a) \Rightarrow \mathsf{out}\neg f), \bullet(\mathsf{in}\, m \wedge \mathsf{in}\, a \Rightarrow \mathsf{out}\, f)\}$ we expect to derive $\mathsf{out}\neg f$, while if we add $\mathsf{in}\, a$ to Γ we expect to derive $\mathsf{out}\, f$.

The adaptive systems presented above can easily be enhanced so that the negation in $\neg\mathsf{in}\, A$ is interpreted as a default negation which holds whenever

[32] In [19] the authors present an embedding of I/O logics into parametrized logic programming that overcomes this difficulty. See Section 5.2 for a comparison with our approach.

there is no input to the contrary. A simple way to achieve this is to use the set of abnormalities $\Omega_{\rm defneg} = \{\mathsf{in}\,A \mid A \in \mathcal{W}\}$. Various techniques are available that allow to define adaptive logics in which both sets of abnormalities $\Omega_{\rm defneg}$ and $\Omega^+_\bullet$ are considered (see e.g., [48,49]). The most straightforward way is to use an adaptive logic based on $\Omega^\bullet_{\rm defneg} = \Omega_{\rm defneg} \cup \Omega^+_\bullet$. From $\Gamma_{\rm defneg}$ we then can derive $\neg\mathsf{in}\,a$ and $\mathsf{out}\neg f$.

Some examples motivate a further refinement of our approach. Take $\Gamma'_{\rm defneg} = \{\mathsf{in}\,b, \mathsf{in}\,e, \bullet((\mathsf{in}\,e \wedge \neg\mathsf{in}\,a) \Rightarrow \mathsf{out}\,c), \bullet(\mathsf{in}\,b \Rightarrow \mathsf{out}\neg c), \bullet((\mathsf{in}\,b \wedge \neg\mathsf{in}\,a) \Rightarrow \mathsf{out}\,d)\}$. With $\Omega^\bullet_{\rm defneg}$ we have the minimal *Dab*-formula $\mathsf{in}\,a \vee \lightning(\mathsf{in}\,b \Rightarrow \mathsf{out}\neg c) \vee \lightning((\mathsf{in}\,e \wedge \neg\mathsf{in}\,a) \Rightarrow \mathsf{out}\,c)$. This implies that $\mathsf{out}\,d$ is not derivable since it can only be derived on the condition $\{\mathsf{in}\,a, \lightning((\mathsf{in}\,b \wedge \neg\mathsf{in}\,a) \Rightarrow \mathsf{out}\,d)\}$.

Under a more strict reading of 'negation as failure to derive' we want to derive $\neg\mathsf{in}\,A$ iff $\mathsf{in}\,A$ is not derivable (via $\mathbf{CMIO}_{\mathsf{R}}$) from the given premises. For $\Gamma'_{\rm defneg}$ this means that we expect $\neg\mathsf{in}\,a$ to be derivable. Given $\neg\mathsf{in}\,a$, we have a conflict between the two triggered conditionals $(\mathsf{in}\,e \wedge \neg\mathsf{in}\,a) \Rightarrow \mathsf{out}\,c$ and $\mathsf{in}\,b \Rightarrow \mathsf{out}\neg c$. Since $(\mathsf{in}\,b \wedge \neg\mathsf{in}\,a) \Rightarrow \mathsf{out}\,d$ is triggered and not related to the conflict we expect also that $\mathsf{out}\,d$ is derivable. One way to achieve this is to sequentially combine adaptive logics—say $\mathbf{AL_1}$ and $\mathbf{AL_2}$—such that the combined logic $\mathbf{AL}$ has the consequence set: $\mathrm{Cn}_{\mathbf{AL}}(\Gamma) = \mathrm{Cn}_{\mathbf{AL_2}}(\mathrm{Cn}_{\mathbf{AL_1}}(\Gamma))$. By letting $\mathbf{AL_1}$ be based on $\Omega_{\rm defneg}$ we first minimize abnormalities of the form $\mathsf{in}\,A$ and hence interpret the given facts under negation as failure. $\mathbf{AL_2}$ is then based on $\Omega^+_\bullet$ which allows to apply detachment to conditionals 'as much as possible'. Dynamic proof theories for such combinations are defined in [45,47]. Another option is e.g. to use lexicographic adaptive logics [49]. In such adaptive logics, both $\neg\mathsf{in}\,a$ and $\mathsf{out}\,d$ are derivable from $\Gamma'_{\rm defneg}$, while neither $\mathsf{out}\,c$ nor $\mathsf{out}\neg c$ are.

Permissions by default. A similar approach can be used to model *weak* or *negative permissions* [9,31]. The idea is that A is permitted whenever we can deduce that there is no obligation to $\neg A$. Arguably, in many applications this notion of weak permissions is too restrictive and a stronger principle seems more adequate which allows us to derive the permission to bring about A whenever $\mathsf{out}\neg A$ is not derivable (and not just if $\neg\mathsf{out}\neg A$ is derived, see also [9]). This can be modelled by $\neg\mathsf{out}\neg A$ where the negation with the wide scope (the one preceding 'out') is a default negation (i.e., it is interpreted as a 'negation as failure to derive'). As an illustration, take $\Gamma_{\rm perm} = \{\mathsf{in}\,p, \mathsf{in}\,q, \bullet(p \rightarrow s), \bullet(q \rightarrow \neg s), \bullet(q \rightarrow (t \vee r))\}$. Note that we have the minimal *Dab*-formula $\lightning(p \rightarrow s) \vee \lightning(q \rightarrow \neg s)$. This means that $\mathsf{out}(t \vee r)$ is derivable and hence also $\neg\mathsf{out}\neg(t \vee r)$ (the permission to bring about $t \vee r$). Now, were we to interpret negations preceeding 'out' as default negations we would also get, for instance, $\neg\mathsf{out}\neg t$ and $\neg\mathsf{out}\neg r$ which makes t and r permitted. Indeed, we can neither derive the obligation to $\neg t$ nor the obligation to $\neg r$ and hence, according to the notion of *negative* permission both t and r are permitted.

Technically this way of modelling weak permissions is realized by letting $\Omega_{\text{wper}} = \{\mathsf{out}\, A \mid A \in \mathcal{W}\}$ and by using combined adaptive logics that first minimize according to the abnormalities in $\Omega_{\bullet}$ (resp. $\Omega_{\bullet}^{+}$) and then according to the abnormalities in Ω_{wper}.[33]

Permissive norms. In some applications it is useful to have permissive norms as part of the conditional norms in the premise set. An option is to use $\mathsf{in}A \Rightarrow \neg\mathsf{out}\neg B$: given A we have a reason to suppose that $\neg B$ is not obliged resp. that B is permitted.[34]

Since in our lower limit logics $\mathbf{CMIO}_{\mathsf{R}}$ we have modus ponens for $\Rightarrow$, we can derive $\neg\mathsf{out}\neg B$ from $\mathsf{in}\, A$ and $\mathsf{in}\, A \Rightarrow \neg\mathsf{out}\neg B$. Depending on the rules in R we get other properties for conditional permissions. E.g., if out is a **KD**-modality (i.e., in all $\mathbf{CMIO}_{\mathsf{R}}$ variants) we get,

$$\vdash (\mathsf{out}\, A \wedge \neg\mathsf{out}\neg B) \supset \neg\mathsf{out}\neg(A \wedge B)$$

Where, additionally, $\Rightarrow$ supports right-weakening (where $A \vdash B$, $\vdash (C \Rightarrow A) \supset (C \Rightarrow B)$), we also have

$$\text{If } A \vdash B \text{ then } \vdash (\mathsf{in}\, C \Rightarrow \neg\mathsf{out}\neg A) \supset (\mathsf{in}\, C \Rightarrow \neg\mathsf{out}\neg B)$$

Here's a simple example for an adaptive proof with the lower limit logic $\mathbf{CMIO}_{\mathsf{R}}$ (where R contains (SI), (WO) and (AND)), the set of abnormalities $\Omega_{\bullet}^{+}$, the minimal abnormality strategy, and the premise set $\Gamma = \{\mathsf{in}\, p, \mathsf{in}\, q, \bullet(\mathsf{in}\, p \Rightarrow \mathsf{out}\neg s), \bullet(\mathsf{in}\, q \Rightarrow \neg\mathsf{out}\neg s), \bullet(\mathsf{in}\, p \Rightarrow \neg\mathsf{out}\neg t)\}$.

1	$\bullet(\mathsf{in}\, p \Rightarrow \mathsf{out}\neg s)$	PREM	$\emptyset$
2	$\bullet(\mathsf{in}\, q \Rightarrow \neg\mathsf{out}\neg s)$	PREM	$\emptyset$
3	$\bullet(\mathsf{in}\, p \Rightarrow \neg\mathsf{out}\neg t)$	PREM	$\emptyset$
4	$\mathsf{in}\, p$	PREM	$\emptyset$
5	$\mathsf{in}\, q$	PREM	$\emptyset$
✓6	$\mathsf{out}\neg s$	1,4; RC	$\{\lightning(\mathsf{in}\, p \Rightarrow \mathsf{out}\neg s)\}$
✓7	$\neg\mathsf{out}\neg s$	2,5; RC	$\{\lightning(\mathsf{in}\, q \Rightarrow \neg\mathsf{out}\, \neg s)\}$

[33] See the paragraph on exceptions above for more details.

[34] To simplify things, in our discussion of conditional permissions we here only consider the option of using the definition of permissions as known from Standard Deontic Logic where P is $\neg O \neg$ (which translates to $\neg\mathsf{out}\neg$ in our setting). Another option would be to use a dedicated permission operator pout.

8	$\neg\mathsf{out}\neg t$	3,4; RC	$\{\lightning(\mathsf{in}\,p \Rightarrow \neg\mathsf{out}\neg t)\}$
9	$\lightning(\mathsf{in}\,p \Rightarrow \mathsf{out}\neg s) \vee \lightning(\mathsf{in}\,q \Rightarrow \neg\mathsf{out}\neg s)$	1,2,4,5; RU	$\emptyset$

It is easy to see that $\neg\mathsf{out}\neg t$ is finally derivable while neither $\mathsf{out}\neg s$ nor $\neg\mathsf{out}\neg s$ are finally derivable due to the minimal *Dab*-formula at line 9.

An interesting further extension of this embedding of permissions in our approach is to prioritize (conditional) permissions over conflicting conditional obligations. Technically this is straightforward: we first minimize abnormalities in $\Omega_{\text{condPerm}} = \{\lightning A \mid A \text{ is of the form } \mathsf{in}\, B \Rightarrow \neg\mathsf{out}\, C\}$ and only then the abnormalities in $\Omega_{\bullet}$. This way, whenever we have $\mathsf{in}\, A$, $\bullet(\mathsf{in}\, A \Rightarrow \mathsf{out}\, B)$ and $\bullet(\mathsf{in}\, A \Rightarrow \neg\mathsf{out}\neg B)$, we will only be able to detach $\neg\mathsf{out}\neg B$. In our example above, this would lead to the derivability of the permission $\neg\mathsf{out}\neg s$ detached at line 7 in the proof, whereas the obligation $\mathsf{out}\neg s$ at line 6 is not derivable. Prioritizing permissions in this way gives us a concept that is similar to Stolpe's notion of 'permission as derogation' [43]. We return to this point in Section 5.3 where we discuss prioritized norms.

5. Discussion

5.1. Modularity and Variants

Recall that all I/O operations are obtained by varying two parameters: (i) the way the maxfamily is used when generating output (join vs. meet of the outfamily), and (ii) the rules under which the set of generators $\mathcal{G}$ is closed. Both are mirrored by a specific feature of our systems. The adaptive strategy provides the counterpart of (i), whereas the rules mentioned in (ii) are translated literally into inference rules for conditionals of the lower limit logic (in line with Table 2). Hence, structural properties of the original system are mirrored by structural properties of the system into which it is translated. This makes our translation procedure very unifying and natural. As a result, one may alter these frameworks in various straightforward ways, while preserving the representation theorems.

Below, we briefly discuss additional variations with respect to (i) and (ii), showing how these are translated into our adaptive framework. We will indicate why these variations seem interesting, but leave their full exploration for a later occasion. Additional options for variation will be discussed in Section 5.3.

Free output. Consider the following example: $\mathcal{A} = \{a, b\}, \mathcal{G} = \{(a, c), (b, d), (a \wedge b, e)\}$ and $\mathcal{C} = \{\neg(c \wedge d)\}$. As explained before, both e and $c \vee d$ are in

$\mathrm{out}_1^{\cap}(\mathcal{G}, \mathcal{A}, \mathcal{C})$. However, there is a difference between both outputs in view of the way they are generated from the two maxfamilies $\mathcal{H} = \{(a, c), (a \wedge b, e)\}$ and $\mathcal{H}' = \{(b, d), (a \wedge b, e)\}$.

Using terminology from default logic, conclusions such as $c \vee d$ may be called *floating conclusions* [28]. In our terms, they can be defined as those formulas in $\mathrm{out}_{\mathsf{R}}^{\cap}(\mathcal{G}, \mathcal{A}, \mathcal{C})$ which cannot be generated from the I/O pairs that are shared by every member of $\mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C})$. Note that $c \vee d$ can only be obtained by applying the pairs (a, b), resp. (c, d), neither of which are in $\mathcal{H} \cap \mathcal{H}'$. In contrast, e is obtained by the application of a conditional $(a \wedge b, e)$ which is contained in both $\mathcal{H}$ and $\mathcal{H}'$.

The status of floating conclusions has been the subject of vigorous debate. Following [28], some have argued that we should accept them just as any other non-monotonic consequence (see also [38]), whereas others have come up with various examples in order to show they are problematic [23]. Consequently, several non-monotonic logics have been modified in order to allow or disallow the derivation of floating conclusions (e.g., [23, 37]).

It is not our aim to take a stance in this long-lasting debate. For the present purposes, note that there is a straightforward alternative to Definition 6 in view of which floating conclusions are avoided[35]:

Definition 17. (*Free output*)

$$\mathrm{out}_{\mathsf{R}}^{\Cap}(\mathcal{G}, \mathcal{A}, \mathcal{C}) =_{\mathsf{df}} \mathrm{out}_{\mathsf{R}} \left(\bigcap \mathsf{maxfamily}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}, \mathcal{C}), \mathcal{A} \right)$$

Our nomenclature echoes that from [40] where a similar technique is applied in a more narrow setting.[36] In the above example, we have $e \in \mathrm{out}_1^{\Cap}(\mathcal{G}, \mathcal{A}, \mathcal{C})$ and $c \vee d \notin \mathrm{out}_1^{\Cap}(\mathcal{G}, \mathcal{A}, \mathcal{C})$.

The operation of free output can be characterized in our adaptive framework by using the so-called *reliability* strategy. Let us illustrate the basic idea behind this strategy in terms of the above example. We take another look at Γ_5 and show how $\mathsf{out}e$ can be derived.

1	$\bullet(a \to c)$	PREM	$\emptyset$
2	$\bullet(b \to d)$	PREM	$\emptyset$
3	$\mathsf{in}a$	PREM	$\emptyset$
4	$\mathsf{in}b$	PREM	$\emptyset$

[35] This proposal is completely analogous to the one from [23, Chap. 7], where a default logic is proposed that invalidates floating conclusions.

[36] More precisely, the *Free Rescher-Manor consequence relation* from [40] reduces to the specific case where $\mathcal{A} = \mathcal{C} = \emptyset$ and all members of $\mathcal{G}$ are of the form $(\top, A)$. We leave the verification of this claim to the interested reader.

5	$\mathsf{con}\neg(c \wedge d)$	PREM	$\emptyset$
6	$\mathsf{out}c$	1,3; RC	$\{\lightning(a \to c)\}\checkmark^{14}$
7	$\mathsf{out}d$	2,4; RC	$\{\lightning(b \to d)\}\checkmark^{14}$
8	$\mathsf{out}(c \wedge d)$	6,7; RU	$\{\lightning(a \to c), \lightning(b \to d)\}\checkmark^{14}$
9	$\lightning(a \to c) \vee \lightning(b \to d)$	1–5; RC	$\emptyset$
10	$\mathsf{out}(c \vee d)$	6; RU	$\{\lightning(a \to c)\}\checkmark^{14}$
11	$\mathsf{out}(c \vee d)$	7; RU	$\{\lightning(b \to d)\}\checkmark^{14}$
12	$\bullet((a \wedge b) \to e)$	PREM	$\emptyset$
13	$\mathsf{in}(a \wedge b)$	3,4;RU	$\emptyset$
14	$\mathsf{out}e$	12,13; RC	$\{\lightning((a \wedge b) \to e)\}$

The idea behind marking for the reliability strategy is straightforward: a line is marked at stage s iff some member of its condition occurs in a minimal *Dab*-formula at stage s. Formally:

DEFINITION 18. Where $Dab(\Delta_1), Dab(\Delta_2), \ldots$ are the minimal *Dab*-formulas derived at stage s, $U_s(\Gamma) = \Delta_1 \cup \Delta_2 \cup \ldots$ is the set of formulas that are *unreliable* at stage s.

DEFINITION 19. A line l with condition Δ is marked at stage s iff $\Delta \cap U_s(\Gamma) \neq \emptyset$.

Using this strategy, not only lines 6–8 are marked at stage 14 (as was the case with the minimal abnormality strategy), but also line 10 and line 11. However, line 14 is not marked.

Where $\dagger \in \{+, -\}$, let $\mathbf{MIO}_{\mathsf{R}}^{\dagger,\Cap}$ be the AL defined by (i) the lower limit logic $\mathbf{CMIO}_{\mathsf{R}}^{\dagger}$, (ii) the set of abnormalities $\Omega_{\bullet} = \{\lightning(A \to B) \mid A, B \in \mathcal{W}\}$, and (iii) the strategy Reliability. We write $\mathbf{MIO}_{\mathsf{R}}^{\Cap}$ to refer to any of the two variants. As before, the adaptive proofs are dynamic. So we need to apply Definitions 12 and 13 to obtain a stable consequence relation for the logics $\mathbf{MIO}_{\mathsf{R}}^{\Cap}$. The following holds[37]:

THEOREM 8. *Where $\mathcal{A}$ is* $\mathbf{CL}$*-consistent or* (F) $\in \hat{\mathsf{R}}$*:*

1. $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{MIO}_{\mathsf{R}}^{-,\Cap}} \mathsf{out}A$ *iff* $A \in \mathrm{out}_{\mathsf{R}}^{\Cap}(\mathcal{G}, \mathcal{A}, \mathcal{C})$
2. *where* (*WO*), (*Z*) $\in \hat{\mathsf{R}}$, $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{MIO}_{\mathsf{R}}^{+,\Cap}} \mathsf{out}A$ *iff* $A \in \mathrm{out}_{\mathsf{R}}^{\Cap}(\mathcal{G}, \mathcal{A}, \mathcal{C})$

Let us make one last remark before leaving the matter. Recall that the strategy only affects the marking definition, whereas the generic inference

[37] See Appendix 4 for the proof.

rules are independent of it. Hence, simply by considering different marking definitions for an adaptive proof one can switch between the various strategies to check whether (according to our insights at the present stage of the proof) some conclusion is e.g. a member of the free constrained output, or whether it only follows by the full meet constrained output.

Other rules for conditionals. Given the existing variation in terms of rules for I/O pairs, one may also consider additional rule systems for conditionals. Such systems can be obtained by skipping some of the rules or by adding others. Makinson and van der Torre make a similar remark in Section 9 of [29], where they discuss several other rules, e.g. contraposition (CONT), (full) transitivity (T), and conditionalisation (COND)[38]:

$$\text{If } (A, B),\ \text{then } (\neg B, \neg A). \tag{CONT}$$

$$\text{If } (A, B) \text{ and } (B, C),\ \text{then } (A, C). \tag{T}$$

$$\text{If } (A, B) \text{ then } (\top, A \supset B). \tag{COND}$$

Both in the unconstrained and the constrained case, our modal framework can easily deal with the addition of such extra rules. That is, the representation theorems mentioned in preceding sections can be shown to hold for arbitrary R as long as R is normal and contains only rules of the form: "If $A_1 \vdash_{\mathbf{CL}} B_1, \ldots, A_n \vdash_{\mathbf{CL}} B_n$ and $(C_1, D_1), \ldots, (C_m, D_m)$, then (E, F)". The rules in R are translated into rules for conditionals in the associated lower limit logic $\mathbf{CMIO}_{\mathsf{R}}$. Thus, we can characterize the operations $\mathrm{out}_{\mathsf{R}}^{\cap}$, $\mathrm{out}_{\mathsf{R}}^{\cup}$ and $\mathrm{out}_{\mathsf{R}}^{\Cap}$ in terms of adaptive proof theories.

5.2. Comparison

Besides our characterization of constrained I/O logics there have been other reconstructions in terms of non-monotonic logics. Already in [29] a link was established between extensions of Reiter's default logic [39] and the outfamilies of out_3^+: the former form a (usually) strict subset of the latter (given an appropriate translation).

In [19] we find a stronger embedding of constrained I/O logics based on out_1^+ and out_3^+, this time into parametrized logic programming. Similar to the lower limit logic of adaptive logics, parametrized logic programs are built on top of a Tarskian core logic $\mathbf{L}$ (this time called the *parameter logic*).

[38] Note that (T) follows from (CT) and (SI).

A (normal) **L**-parametrized logic program consists of a set of rules of the form[39]:

$$C \leftarrow A_1, \ldots, A_n, \mathsf{not}\, B_1, \ldots, \mathsf{not}\, B_m \tag{1}$$

where C, A_i, B_j ($1 \leq i \leq n$, $1 \leq j \leq m$) are **L**-formulas and not is interpreted in terms of 'negation as failure'.[40] For instance, out_3^+ is characterized by the following logic program based on $\langle \mathcal{G}, \mathcal{A} \rangle$ (disregarding constraints), where the language of **CL** is enriched by a set Φ of auxiliary constants r_Φ for every $r = (A, B)$ in $\mathcal{G}$ encoding that r is 'discharged'[41]:

$$\mathcal{P}_3 = \big\{ B \leftarrow A, \mathsf{not}\, r_\Phi \mid r = (A, B) \in \mathcal{G} \big\} \cup \big\{ A \leftarrow \mid A \in \mathcal{A} \big\} \cup \\ \big\{ r_\Phi \leftarrow A, \mathsf{not}\, B \mid r = (A, B) \in \mathcal{G} \big\}$$

The idea is to select stable models of $\mathcal{P}_3$ that verify formulas r_Φ (where $r = (A, B) \in \mathcal{G}$) 'as little as possible'. This in turn makes sure that detachment is applied to I/O pairs (A, B) as much as possible (in view of $B \leftarrow A, \mathsf{not}\, r_\Phi$).

Hence, the basic idea is rather similar to our characterization in terms of adaptive logics. In both characterizations it is warranted that detachment is applied to as many I/O pairs (A, B) as possible: in the characterization by ALs this is done by avoiding abnormalities of the type $\bullet(A \to B) \wedge \neg(A \to B)$, in the characterization in terms of logic programs this is taken care of by avoiding formulas of the type r_Φ (where $r = (A, B) \in \mathcal{G}$). Both representations provide a characterization of the maxfamilies: in the representation by

---

[39]The structural similarities between parametrized logic programming and ALs clearly motivate future investigations concerning comparisons between the two frameworks and possible embeddings. E.g., we know that default and autoepistemic logic can be embedded in ALs on the basis of the modal characterization in [26]. This motivates modal translations of the rules of parametrized logic programming. E.g., a possible path is to translate (1) to $(\Box A_1 \wedge \ldots \wedge \Box A_n \wedge \neg\Box B_1 \wedge \ldots \wedge \neg\Box B_m) \supset \Box C$ for a suitable normal $\Box$-operator. Negation as failure may then be implemented by treating $\Box A$ as an abnormality. In order to embed ALs with the minimal abnormality strategy in parametrized logic programs rules such as $\neg A \leftarrow \mathsf{not}\, A$ (for all abnormalities A) may be used for implementing the idea that abnormalities are classically negated as little as possible.

[40]The essential difference with orthodox logic programs is that parametrized logic programs allow for complex formulas in the language of **L** in the head and in the body of the rules.

[41]The original characterization in [19, Theorem 1.2] contained a mistake. In correspondence with the authors we present here a corrected version. Compared with the original characterization we have two main changes: 1. there is no need for the additional $\overline{\cdot}$-operator, instead constants of the type r_Φ are introduced to encode that a rule is not applied; 2. in addition to selecting the stable models of the program $\mathcal{P}_3$, a post-selection is needed that selects all stable models that validate minimally many formulas in Φ (the idea is to apply detachment to as many I/O pairs as possible). A similar correction is needed in order to obtain a correct characterization of out_1^+ in [19, Theorem 1.1].

means of ALs the minimal choice sets correspond exactly to the maxfamilies in view of Theorem 5, in the representation by means of logic programs a subset of the stable models corresponds exactly to the maxfamilies (see [19, Theorem 1] and Fn. 41).

5.3. Outlook

Just like I/O logics, our adaptive characterizations can easily be varied, enriched, and adjusted for different application contexts. This has already been explicated for some straightforward variants in Section 5.1. Let us in this outlook give some examples of how the presented framework can be further enhanced by enriching the formal language or changing the underlying monotonic logic.

Priorities. In some applications it may be useful to introduce priorities among the I/O pairs. This idea is not new in the context of logics based on maximal consistent subsets. For instance, in Brewka's 'preferred subtheories'-approach [15] priorities among formulas are considered when the maximal consistent subsets are selected that form the basis for his consequence relations. A similar procedure can be realized by a slight generalization of the framework presented in this paper. Instead of modeling all given I/O conditionals on a par by means of $\bullet$, we encode priorities among them by prefixing them with $\bullet_i$ ($i \in \mathbb{N}$) where i indicates the priority we attach to the conditional. The abnormalities are then presented by the set $\Omega = \{\bullet_i A \wedge \neg A \mid i \geq 1, A \in \mathcal{W}'\}$. Moreover, we can make use of so-called lexicographic ALs [49,50] that take care of the priorities in a natural way.

Instead of 'hard-coding' priorities one may also consider priorities that arise in view of logical relationships among norms such as specificity cases in which more specific norms override conflicting norms (see e.g. [44,52]). One challenge is that specificity introduces an additional level of defeasibility, since norms are in force unless they are canceled by more specific norms with which they conflict. This may also motivate giving up on (SI) (or at least to restrict it) which is currently not possible in our systems.[42]

Quantitative considerations. In other applications also quantitative considerations may play a role: if we have two maxfamilies to choose from, but

[42] A systematic investigation into a particular case of specificity in the context of I/O logic is provided in Stolpe's [43] where permission is thought of as derogation. See also the paragraph on permissions in Section 4.

in one (significantly) more I/O pairs are 'violated', we may have good reasons to choose the option to violate fewer norms. A format for ALs that is expressive enough to embed also quantitative variants has been recently investigated in [45, Chap. 5]. In view of these results it is easy to devise variants of the logics presented in this paper that implement quantitative considerations.

Inconsistent facts. I/O logics only deal with conflicts among the I/O pairs but are not designed for applications where also the factual input may be conflicting. This can easily be fixed in a way that is coherent with one of the main mechanisms behind I/O logics: namely to work with maximal consistent families. The idea is simply to first form maximal consistent subsets of the factual input and subsequently to apply to each of these sets the respective I/O function. Both steps can straightforwardly be integrated in an AL framework that generalizes the framework presented in this paper by means of combining it with the techniques presented in [34].

Going predicative. Using our framework opens the prospect of applying the constrained I/O mechanism to predicate logic: i.e., for instance to allow for input of the form $\mathsf{in}(\forall x P(x))$ or conditionals of the form $(\forall x P(x)) \rightarrow (\exists x Q(x))$ or $\forall x(P(x) \rightarrow Q(x))$. The reason why the original (constrained) I/O functions are suboptimal for the explication of defeasible reasoning in this context is due to the undecidability of predicate logic: there is no effective way to perform the consistency check that is necessary to calculate the maxfamilies.[43] The dynamic proof theory of ALs comes in handy since it doesn't force us to make consistency checks 'on the spot'. Instead, we can defeasibly assume that a set of conditionals is consistent. E.g., in a $\mathbf{MIO}^{\cup}_{\mathsf{R}}$-proof from $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}}$ we may derive $\mathsf{out}A$ on the condition $\Delta = \{\lightning(B \rightarrow C) \mid (B, C) \in \mathcal{H}\}$ where $\mathcal{H} \subseteq \mathcal{G}$. This means that we derive $\mathsf{out}A$ on the assumption that $\mathcal{H}$ is consistent with the given input and the given constraints (i.e., it is a subset of a maxfamily of $\langle \mathcal{G}, \mathcal{A}, \mathcal{C} \rangle$). In case this doesn't hold we know that we will be able to derive $Dab(\Delta)$ on the empty condition eventually and hence we will be forced to mark the line.[44]

[43] A similar observation has been made by Horty [21, p. 50] in the context of his dyadic enrichment of an older system by Van Fraassen [53] that is based on consistency considerations.

[44] The merits of the dynamic proofs of ALs when a positive test for consistency is not available have been pointed out before, e.g. by Batens in the context of ALs for inductive generalizations [4].

6. Conclusion

We reconstructed I/O logics as modal adaptive logics and proved this reconstruction to be equivalent to the original definition. Apart from the purely technical interest in translating systems from one formal framework to another, we argued that our characterization has some additional advantages.

First, our logics come with a proof theory that mirrors various types of defeasible reasoning. Both of Pollock's notions of synchronic and diachronic defeasibility are modeled in adaptive proofs. Moreover, the proof theory can deal with the case-specific retraction of conclusions in the light of new input (Section 3.1).

Second, our reconstruction allows for more flexibility in the formulation of the premises. E.g., in order to express violations and sanctions deontic information may appear in the body of our conditionals that express I/O rules (Section 4).

Third, making use of techniques from the adaptive logics framework, we obtain new and interesting variations of the original I/O functions. This can be achieved by varying either the adaptive strategy or by varying the rules of the lower limit logic (Section 5.1).

Acknowledgements. The contributions of Christian Straßer and Mathieu Beirlaen were supported by a Sofja Kovalevskaja award of the Alexander von Humboldt Foundation, funded by the German Ministry for Education and Research. In addition, the contribution of Mathieu Beirlaen was supported by the project *Logics of discovery, heuristics and creativity in the sciences* (PAPIIT, IN400514) granted by the National Autonomous University of Mexico (UNAM), and by the *Programa de Becas Posdoctorales* of the *Coordinación de Humanidades* at UNAM. The contribution of Frederik Van De Putte was supported by subventions of the Flemish Fund for Scientific Research (FWO—Vlaanderen).

Appendices

In this appendix we assume a fixed normal set of rules R for I/O-pairs (i.e., (SI), (EQ), (AND) $\in \hat{\mathsf{R}}$). Hence we do not restrict ourselves to the rules that are considered in Section 2.1. The considered rules have one of the two forms of the left column of Table 2 and are translated as described there.

We use the following notations. Where $\Gamma \subseteq \mathcal{W}$ and $\pi \in \{\mathsf{in}, \mathsf{out}, \mathsf{con}\}$, $\Gamma^\pi = \{\pi A \mid A \in \Gamma\}$. Where $\mathcal{G}$ is a set of I/O-pairs, $\mathcal{G}^{\rightarrow} = \{A \rightarrow B \mid (A, B) \in \mathcal{G}\}$.

Appendix 1: Proof of Theorem 2

The proofs of the following two facts follow immediately by the way the rules are translated in Table 2.

FACT 5. $(A, B) \in \mathcal{G}_{\mathsf{R}}$ *iff* $\mathcal{G}^{\rightarrow} \vdash_{\mathbf{MIO}_{\mathsf{R}}} A \rightarrow B$.

FACT 6. *If* $\mathcal{G}^{\rightarrow} \vdash_{\mathbf{MIO}_{\mathsf{R}}} A \Rightarrow B$ *then there are* A' *and* B' *such that* $A = \mathsf{in}A'$ *and* $B = \mathsf{out}B'$.

The following is a well-known fact about the modal logic **K**.

FACT 7. $A \in Cn_{\mathbf{CL}}(\mathcal{A})$ *iff* $\mathcal{A}^{\mathsf{in}} \vdash_{\mathbf{MIO}_{\mathsf{R}}} \mathsf{in}A$.

THEOREM 9. *If* $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$*, then* $\Gamma^{\mathcal{G},\mathcal{A}} \vdash_{\mathbf{MIO}_{\mathsf{R}}} \mathsf{out}A$.

PROOF. Suppose $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. So there is a $B \in Cn_{\mathbf{CL}}(\mathcal{A})$ such that $(B, A) \in \mathcal{G}_{\mathsf{R}}$. By Facts 5 and 7 respectively, $\mathcal{G}^{\rightarrow} \vdash_{\mathbf{MIO}_{\mathsf{R}}} B \rightarrow A$ and $\mathcal{A}^{\mathsf{in}} \vdash_{\mathbf{MIO}_{\mathsf{R}}} \mathsf{in}B$. By the monotonicity of $\mathbf{MIO}_{\mathsf{R}}$ and ($\mathrm{MP}_{\Rightarrow}$), $\Gamma^{\mathcal{G},\mathcal{A}} \vdash_{\mathbf{MIO}_{\mathsf{R}}} \mathrm{out}A$. ■

The next fact follows by the definition of $\mathrm{out}_{\mathsf{R}}$, (F) and since in is a **KD**-modality.

FACT 8. *Where* $\mathcal{A}$ *is* **CL**-*inconsistent and (F)* $\in \hat{\mathsf{R}}$*,* $out_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ *is* **CL**-*inconsistent and* $\Gamma^{\mathcal{A},\mathcal{G}} \vdash_{\mathbf{MIO}_{\mathsf{R}}} \perp$.

THEOREM 10. *Where* $\mathcal{A}$ *is* **CL**-*consistent or (F)* $\in \hat{\mathsf{R}}$*,*

1. *If* $\Gamma^{\mathcal{A},\mathcal{G}} \vdash_{\mathbf{MIO}_{\mathsf{R}}^{-}} \mathsf{out}A$*, then* $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$.
2. *Where (WO), (Z)* $\in \hat{\mathsf{R}}$*, if* $\Gamma^{\mathcal{A},\mathcal{G}} \vdash_{\mathbf{MIO}_{\mathsf{R}^{+}}} \mathsf{out}A$*, then* $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$.

PROOF. In case $\mathcal{A}$ is **CL**-inconsistent and (F) $\in \hat{\mathsf{R}}$, the theorem follows by Fact 8. Suppose $\mathcal{A}$ is **CL**-consistent and $A \notin \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. Let $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$, where

$\Gamma_1 = \{\mathsf{in}B \mid B \in Cn_{\mathbf{CL}}(\mathcal{A})\} \cup \{\neg\mathsf{in}C \mid C \in \mathcal{W} - Cn_{\mathbf{CL}}(\mathcal{A})\}$

$\Gamma_2 = \{B \rightarrow C \mid (B, C) \in \mathcal{G}_{\mathsf{R}}\} \cup \{\neg(B \Rightarrow C) \mid$ there are no $B', C' \in \mathcal{W}$ for which $B = \mathsf{in}B', C = \mathsf{out}C'$ and $(B', C') \notin \mathcal{G}_{\mathsf{R}}\}$

$\Gamma_3 = \{\mathsf{out}B \mid B \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})\} \cup \{\neg\mathsf{out}C \mid C \in \mathcal{W} - \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})\}$

We first show that Γ is **CL**-consistent.[45] Since in, out, and $\Rightarrow$ have no meaning in **CL** we only need to show that there are no $B, C \in \mathcal{W}'$ for which $B \Rightarrow C, \neg(B \Rightarrow C) \in \Gamma_2$. By the construction of Γ_2 we have to show that there are no $B, C \in \mathcal{W}$ for which $B \rightarrow C, \neg(B \rightarrow C) \in \Gamma_2$. This follows directly by the construction of Γ_2.

Let Γ' be a maximal **CL**-consistent extension of Γ. Note that (†) $B \supset C \in \Gamma'$ iff $B \notin \Gamma'$ or $C \in \Gamma'$ and (‡) $\neg B \in \Gamma'$ iff $B \notin \Gamma'$. We now prove for all $D \in \mathcal{W}'$:

$$\Gamma' \vdash_{\mathbf{MIO}_{\mathsf{R}^{-}}} D \text{ iff } D \in \Gamma' \tag{2}$$

$$\text{where (WO), (Z)} \in \hat{\mathsf{R}}, \Gamma' \vdash_{\mathbf{MIO}_{\mathsf{R}^{+}}} D \text{ iff } D \in \Gamma' \tag{3}$$

[45] Note that at this point we do not yet establish that Γ is $\mathbf{MIO}_{\mathsf{R}}$-consistent.

To prove (2) (resp. (3)) it suffices to show that *(i)* whenever D is a $\mathbf{MIO}_{\mathsf{R}}{}^{-}$-axiom (resp. a $\mathbf{MIO}_{\mathsf{R}}{}^{+}$-axiom), then $D \in \Gamma'$ and *(ii)* Γ' is closed under all the $\mathbf{MIO}_{\mathsf{R}}{}^{-}$-rules (resp. all the $\mathbf{MIO}_{\mathsf{R}}{}^{+}$-rules).

Ad (i). For the **CL**-axioms, this follows immediately in view of the construction. So we move on to the **KD**-axioms for in.

Ad (Kin). $\mathsf{in}(B \supset C) \supset (\mathsf{in}B \supset \mathsf{in}C) \in \Gamma'$ iff [by (†)] $\mathsf{in}(B \supset C) \notin \Gamma'$ or $\mathsf{in}B \notin \Gamma'$ or $\mathsf{in}C \in \Gamma'$ iff [by (‡)] $\neg\mathsf{in}(B \supset C) \in \Gamma'$ or $\neg\mathsf{in}B \in \Gamma'$ or $\mathsf{in}C \in \Gamma'$ iff [by the construction] $B \supset C \notin Cn_{\mathbf{CL}}(\mathcal{A})$ or $B \notin Cn_{\mathbf{CL}}(\mathcal{A})$ or $C \in Cn_{\mathbf{CL}}(\mathcal{A})$. The latter holds by modus ponens.

Ad (Din). $\mathsf{in}B \supset \neg\mathsf{in}\neg B \in \Gamma'$ iff [by (†)] $\mathsf{in}B \notin \Gamma'$ or $\neg\mathsf{in}\neg B \in \Gamma'$ iff [by (‡)] $\neg\mathsf{in}B \in \Gamma'$ or $\neg\mathsf{in}\neg B \in \Gamma'$ iff [by the construction] $B \notin Cn_{\mathbf{CL}}(\mathcal{A})$ or $\neg B \notin Cn_{\mathbf{CL}}(\mathcal{A})$. The latter holds by the consistency of $\mathcal{A}$.

Ad (NECin). This follows immediately by the construction of Γ_1.

Suppose now (WO), (Z) $\in \hat{\mathsf{R}}$. *Ad* (Kout). Suppose $\mathsf{out}(B \supset C), \mathsf{out}B \in \Gamma'$. Hence, $B \supset C, B \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. By Definition 1, there are $(A, B \supset C), (A', B) \in \mathcal{G}_{\mathsf{R}}$ for some $A, A' \in Cn_{\mathbf{CL}}(\mathcal{A})$. By (SI) and (AND), $(A \wedge A', B \wedge (B \supset C)) \in \mathcal{G}_{\mathsf{R}}$. By (WO), $(A \wedge A', C) \in \mathcal{G}_{\mathsf{R}}$. Hence, $C \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ and thus, $\mathsf{out}C \in \Gamma_3 \subseteq \Gamma'$. By (†), $\mathsf{out}(B \supset C) \supset (\mathsf{out}B \supset \mathsf{out}C) \in \Gamma'$.

Ad (NECout). Suppose $\vdash_{\mathbf{CL}} A$. Hence, by (Z) and (WO), $(\top, A) \in \mathcal{G}_{\mathsf{R}}$. Thus, also $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. By the construction of Γ_3, $\mathsf{out}A \in \Gamma_3 \subseteq \Gamma'$.

Ad (ii). Ad (MP$_{\Rightarrow}$). Let $A, A \Rightarrow B \in \Gamma'$. By the construction of Γ_2, there are $A', B' \in \mathcal{W}$ such that $A = \mathsf{in}A'$, $B = \mathsf{out}B'$, and $(A', B') \in \mathcal{G}_{\mathsf{R}}$. By the construction of Γ_1, $A' \in Cn_{\mathbf{CL}}(\mathcal{A})$. Hence, by Definition 1, $B' \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. By the construction of Γ_3, $\mathsf{out}B' \in \Gamma_3$ and hence $\mathsf{out}B \in \Gamma'$.

As for the rules in $\mathsf{R}^{\rightarrow}$ this follows immediately by the translation schemes in Table 2, the construction of Γ_2, and Facts 5 and 6.

By (2) (resp. (3)), $\Gamma' \nvdash_{\mathbf{MIO}_{\mathsf{R}}} \bot$ since $\bot \notin \Gamma'$. Hence since $\neg\mathsf{out}A \in \Gamma_3 \subseteq \Gamma'$, $\mathsf{out}A \notin \Gamma'$ and thus, by (2) (resp. (3)), $\Gamma' \nvdash_{\mathbf{MIO}_{\mathsf{R}}} \mathsf{out}A$. Note that $\Gamma^{\mathcal{G},\mathcal{A}} \subseteq \Gamma'$. By the monotonicity of $\mathbf{MIO}_{\mathsf{R}}$, $\Gamma^{\mathcal{G},\mathcal{A}} \nvdash_{\mathbf{MIO}_{\mathsf{R}}} \mathsf{out}A$. ■

Theorem 2 is an immediate consequence of Theorems 9 and 10.

Appendix 2: Some Properties of $\mathbf{CMIO}_{\mathsf{R}}$

In this section, we prove some properties of the systems $\mathbf{CMIO}_{\mathsf{R}}$ which will be helpful for establishing the representation theorems in the next two sections. The properties and their proofs are strongly linked to those from the previous section.

Notation: To avoid clutter, we shall use $\vdash$ ($\vdash^{-}, \vdash^{+}$) to abbreviate $\vdash_{\mathbf{CMIO}_{\mathsf{R}}}$ ($\vdash_{\mathbf{CMIO}_{\mathsf{R}}{}^{-}}$, $\vdash_{\mathbf{CMIO}_{\mathsf{R}}{}^{+}}$) in the remainder of this appendix.

By Theorem 9 and since $\mathbf{CMIO}_{\mathsf{R}}$ is a monotonic extension of $\mathbf{MIO}_{\mathsf{R}}$, we have:

Corollary 2. *Where $\mathcal{C} \subseteq \mathcal{W}$: If $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$, then $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash \mathsf{out}A$.*

Fact 9. *$C \in Cn_{\mathbf{CL}}(\mathcal{C})$ iff $\mathcal{C}^{\mathsf{con}} \vdash \mathsf{con}C$.*

THEOREM 11. *Where $\mathcal{A}$ is consistent or (F) $\in \hat{\mathsf{R}}$:*

1. $\text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) \cup \mathcal{C} \vdash_{\mathbf{CL}} \bot$ *iff* $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash^{-} \bot$.
2. *Where (WO), (Z) $\in \hat{\mathsf{R}}$,* $\text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) \cup \mathcal{C} \vdash_{\mathbf{CL}} \bot$ *iff* $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash^{+} \bot$.

PROOF. The case where $\mathcal{A}$ is **CL**-inconsistent and (F) $\in \hat{\mathsf{R}}$ is trivial (see also Fact 8). Suppose thus that $\mathcal{A}$ is **CL**-consistent.

($\Rightarrow$) Suppose $\text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ is **CL** inconsistent with $\mathcal{C}$. By Remark 1, there is an A for which $\mathcal{C} \vdash_{\mathbf{CL}} A$ and $\neg A \in \text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. By Corollary 2, $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash \mathsf{out}\neg A$. By Fact 9 and the monotonicity of $\mathbf{CMIO}_{\mathsf{R}}$, $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash \mathsf{con}A$ and thus by (ROC), $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash \neg\mathsf{out}\neg A$. Altogether, $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash \bot$.

($\Leftarrow$) Suppose $\text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) \cup \mathcal{C} \nvdash_{\mathbf{CL}} \bot$. Let $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$, where Γ_1, Γ_2 and Γ_3 are defined as in the proof for Theorem 10, and $\Gamma_4 = \{\mathsf{con}A \mid A \in Cn_{\mathbf{CL}}(\mathcal{C})\} \cup \{\neg\mathsf{con}B \mid B \in \mathcal{W} - Cn_{\mathbf{CL}}(\mathcal{C})\}$.

By an analogous argument to the one in Theorem 10 we can show that Γ is **CL**-consistent. Let Γ' be a maximal **CL**-consistent extension of Γ. We have:

$$\Gamma' \vdash^{-} A \text{ iff } A \in \Gamma' \tag{4}$$

$$\text{where (WO), (Z)} \in \hat{\mathsf{R}}, \Gamma' \vdash^{+} A \text{ iff } A \in \Gamma' \tag{5}$$

This can be shown by an analogous argument to the argument for (2) (resp. (3)) in Theorem 10. For (Kcon) and (Dcon) we proceed analogously as for (Kin) and (Din). For the axiom (ROC), we have: $\mathsf{con}A \supset \neg\mathsf{out}\neg A \in \Gamma'$ iff [by (†)] $\mathsf{con}A \notin \Gamma'$ or $\neg\mathsf{out}\neg A \in \Gamma'$ iff [by (‡) and the construction] $A \notin Cn_{\mathbf{CL}}(\mathcal{C})$ or $\neg A \notin \text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. The latter holds in view of the supposition and Remark 1.

By the construction, $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \subseteq \Gamma'$. By the monotonicity of $\mathbf{CMIO}_{\mathsf{R}}{}^{-}$ (resp. of $\mathbf{CMIO}_{\mathsf{R}}{}^{+}$), (4) (resp. (5)), and the $\mathbf{CMIO}_{\mathsf{R}}$-consistency of Γ', $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \nvdash \bot$. ■

THEOREM 12. *Where* $\text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) \cup \mathcal{C}$ *is* **CL**-*consistent and ($\mathcal{A}$ is* **CL**-*consistent or (F) $\in \hat{\mathsf{R}}$):*

1. $A \in \text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ *iff* $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash^{-} \text{out}A$.
2. *Where (WO), (Z) $\in \hat{\mathsf{R}}$,* $A \in \text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$ *iff* $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash^{+} \text{out}A$.

PROOF. ($\Rightarrow$) Immediate in view of Corollary 2. ($\Leftarrow$) Suppose $A \notin \text{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A})$. Let Γ' be constructed as in the proof of Theorem 11. By the construction and the supposition, $\neg\mathsf{out}A \in \Gamma'$. By the consistency of Γ', $\mathsf{out}A \notin \Gamma'$. Since $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \subseteq \Gamma'$, by the monotonicity of $\mathbf{CMIO}_{\mathsf{R}}$, (4), $\Gamma^{\mathcal{G},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \nvdash \mathsf{out}A$. ■

Appendix 3: Proof of Theorem 3

We refer to Section 3.2 for the outline of the proof. As shown there, it suffices to prove Lemma 2, Theorems 5 and 6, which we shall do here.

Preliminaries. Let $\mathcal{W}^{\not\bullet}$ denote the set of all formulas in $\mathcal{W}^c$ without occurrences of $\bullet$, and $\mathcal{W}^{\rightarrow} = \{A \rightarrow B \mid A, B \in \mathcal{W}\}$. Where $\Delta \subseteq \mathcal{W}^{\rightarrow}$, let $\Delta^{\bullet} = \{\bullet A \mid A \in \Delta\}$. Recall that $\Omega_{\bullet}(\Delta) = \{\lightning(A \rightarrow B) \mid A \rightarrow B \in \Delta\}$ and, where $\mathcal{G}$ is a set of I/O-pairs, $\Omega_{\bullet}(\mathcal{G})$ is an abbreviation for $\Omega_{\bullet}(\mathcal{G}^{\rightarrow})$. We will in the remainder sometimes use the inverse function $\mho$, which is defined as follows: where $\Theta \subseteq \Omega_{\bullet}$, $\mho(\Theta) = \{A \rightarrow B \mid \lightning(A \rightarrow B) \in \Theta\}$.

Proof of Lemma 2

Since $\bullet$ is a dummy operator in $\mathbf{CMIO}_{\mathsf{R}}$, we have:

FACT 10. *Where $\Gamma \subseteq \mathcal{W}^{\not\bullet}$, $\Delta \cup \{A\} \subseteq \mathcal{W}^{\rightarrow}$ and $B \in \mathcal{W}^c$: $\Gamma \cup \Delta^{\bullet} \vdash B \vee \bullet A$ iff ($\Gamma \cup \Delta^{\bullet} \vdash B$ or $A \in \Delta$).*

LEMMA 3. *Where $\Gamma \subseteq \mathcal{W}^{\not\bullet}$, $\Delta \subseteq \mathcal{W}^{\rightarrow}$, $\Theta \subseteq_f \Omega_{\bullet}$ and $A \in \mathcal{W}^c$: if $\Gamma \cup \Delta^{\bullet} \vdash A \vee Dab(\Theta)$, then $\Gamma \cup \Delta^{\bullet} \vdash A \vee Dab(\Theta \cap \Omega_{\bullet}(\Delta))$.*

PROOF. Suppose $\Gamma \cup \Delta^{\bullet} \vdash A \vee Dab(\Theta)$. Let $\Theta - \Omega_{\bullet}(\Delta) = \{\bullet A_1 \wedge \neg A_1, \ldots, \bullet A_n \wedge \neg A_n\}$. Hence, $(\star)$ $A_i \notin \Delta$ where $1 \leq i \leq n$.

By the supposition and **CL**, $\Gamma \cup \Delta^{\bullet} \vdash A \vee Dab(\Theta - \{\bullet A_1 \wedge \neg A_1\}) \vee \bullet A_1$. By $(\star)$ and Fact 10, $\Gamma \cup \Delta^{\bullet} \vdash A \vee Dab(\Theta - \{\bullet A_1 \wedge \neg A_1\})$. Repeating the same reasoning for A_2, ..., and A_n, we can derive that $\Gamma \cup \Delta^{\bullet} \vdash A \vee Dab(\Theta - (\Theta - \Omega_{\bullet}(\Delta)))$, and hence $\Gamma \cup \Delta^{\bullet} \vdash A \vee Dab(\Theta \cap \Omega_{\bullet}(\Delta))$. ■

Lemma 2 is a special case of Lemma 3 where $\Gamma = \mathcal{A}^{\mathsf{in}} \cup \mathcal{C}^{\mathsf{con}}$ and $\Delta = \mathcal{G}^{\rightarrow}$.

Proof of Theorem 5

Since $\bullet$ is a dummy operator in $\mathbf{CMIO}_{\mathsf{R}}$, we have:

FACT 11. *Where $\Gamma \cup \{A\} \subseteq \mathcal{W}^{\not\bullet}$ and $\Delta \subseteq \mathcal{W}^{\rightarrow}$: $\Gamma \cup \Delta^{\bullet} \vdash A$ iff $\Gamma \vdash A$.*

LEMMA 4. *Where $\Gamma \subseteq \mathcal{W}^{\not\bullet}$ and $\Delta \subseteq \mathcal{W}^{\rightarrow}$: (i) if $\Theta \in \Sigma(\Gamma \cup \Delta^{\bullet})$ and $\Gamma \nvdash \bot$, then $\mho(\Theta) \subseteq \Delta$, (ii) if $\Theta \in \Phi(\Gamma \cup \Delta^{\bullet})$ and $\Gamma \nvdash \bot$, then $\mho(\Theta) \subseteq \Delta$.*

PROOF. Ad (i). Suppose the antecedent holds. Hence $\Gamma \cup \Delta^{\bullet} \vdash \bigvee \Theta$. By addition, $\Gamma \cup \Delta^{\bullet} \vdash \bot \vee \bigvee \Theta$. By Lemma 3, $\Gamma \cup \Delta^{\bullet} \vdash \bot \vee \bigvee(\Theta \cap \Omega_{\bullet}(\Delta))$. Assume first that $\Theta \cap \Omega_{\bullet}(\Delta) = \emptyset$. In that case $\Gamma \cup \Delta^{\bullet} \vdash \bot$ and hence by Fact 11 also $\Gamma \vdash \bot$,—a contradiction. Thus, $\Theta \cap \Omega_{\bullet}(\Delta) \neq \emptyset$ and $\Gamma \cup \Delta^{\bullet} \vdash \bigvee(\Theta \cup \Omega_{\bullet}(\Delta))$. Were $\Theta - \Omega_{\bullet}(\Delta) \subset \Theta$ then $\bigvee \Theta$ would not be a minimal *Dab*-consequence. Thus, $\Theta \cap \Omega_{\bullet}(\Delta) = \Theta$ and hence $\mho(\Theta) \subseteq \Delta$. Ad (ii). This follows immediately by (i) since for each $\Theta \in \Phi(\Gamma \cup \Delta^{\bullet})$, $\Theta \subseteq \bigcup \Sigma(\Gamma \cup \Delta^{\bullet})$. ■

THEOREM 13. *Where $\Gamma \subseteq \mathcal{W}^{\not\bullet}$, $\Delta \subseteq \mathcal{W}^{\rightarrow}$, $\Theta \subseteq \Omega_{\bullet}$ and $\Gamma \nvdash \bot$: Θ is a choice set of $\Sigma(\Gamma \cup \Delta^{\bullet})$ iff $\Gamma \cup (\Delta - \mho(\Theta)) \nvdash \bot$.*

Proof. ($\Rightarrow$) Suppose $\Gamma \cup (\Delta - \mho(\Theta)) \vdash \bot$. By compactness and the deduction theorem, there is a finite $\Lambda \subseteq \Delta - \mho(\Theta)$ such that $\Gamma \vdash \neg \bigwedge \Lambda$. Hence also $\Gamma \cup \Delta^{\bullet} \vdash \neg \bigwedge \Lambda$. Note that, since $\Lambda \subseteq \Delta$, also $\Gamma \cup \Delta^{\bullet} \vdash \bullet A$ for all $A \in \Lambda$. Hence, $\Gamma \cup \Delta^{\bullet} \vdash \bigvee \Omega_{\bullet}(\Lambda)$. It follows that there is a $\Lambda' \subseteq \Lambda$ with $\Omega_{\bullet}(\Lambda') \in \Sigma(\Gamma \cup \Delta^{\bullet})$. Note that also $\Omega_{\bullet}(\Lambda') \cap \Theta = \emptyset$. Hence Θ is not a choice set of $\Sigma(\Gamma \cup \Delta^{\bullet})$.

($\Leftarrow$) Suppose Θ is not a choice set of $\Sigma(\Gamma \cup \Delta^{\bullet})$. Let $\Lambda \in \Sigma(\Gamma \cup \Delta^{\bullet})$ be such that $\Theta \cap \Lambda = \emptyset$. By Lemma 4, we have two cases: (1) $\Gamma \vdash \bot$, or (2) $\mho(\Lambda) \subseteq \Delta$.

Ad (1): By the monotonicity of $\mathbf{CMIO}_{\mathsf{R}}$, $\Gamma \cup (\Delta - \mho(\Theta)) \vdash \bot$.

Ad (2): Since $\Lambda \in \Sigma(\Gamma \cup \Delta^{\bullet})$, $\Gamma \cup \Delta^{\bullet} \vdash \bigvee \Lambda$. Hence also $\Gamma \cup \Delta^{\bullet} \vdash \bigvee\{\neg A \mid A \in \mho(\Lambda)\}$. By Fact 11, $\Gamma \vdash \bigvee\{\neg A \mid A \in \mho(\Lambda)\}$, and hence ($\star$) $\Gamma \cup \mho(\Lambda) \vdash \bot$. Since $\Lambda \cap \Theta = \emptyset$, $\mho(\Lambda) \subseteq \Delta - \mho(\Theta)$. Hence by ($\star$) and the monotonicity of $\mathbf{CMIO}_{\mathsf{R}}$, $\Gamma \cup (\Delta - \mho(\Theta)) \vdash \bot$. ■

Theorem 14. *Let $\Gamma \subseteq \mathcal{W}^{\not\bullet}$, $\Delta \subseteq \mathcal{W}^{\rightarrow}$, $\Theta \subseteq \Omega_{\bullet}$ and $\Gamma \not\vdash \bot$. We have:*

$$\Theta \in \Phi(\Gamma \cup \Delta^{\bullet}) \textit{ iff } \Delta - \mho(\Theta) \in \mathbb{M}(\Gamma, \Delta),$$

where $\mathbb{M}(\Gamma, \Delta)$ is the set of all $\subset$-maximal $\Delta' \subseteq \Delta$ such that $\Gamma \cup \Delta' \nvdash \bot$.

Proof. ($\Rightarrow$) Suppose $\Theta \in \Phi(\Gamma \cup \Delta^{\bullet})$. Assume $\Delta - \mho(\Theta) \notin \mathbb{M}(\Gamma, \Delta)$. Hence, by Theorem 13 there is a $\Lambda \subseteq \Delta$ such that $\Lambda \supset \Delta - \mho(\Theta)$ and $\Gamma \cup \Lambda \nvdash \bot$. Let $\Theta' = \Omega_{\bullet}(\Delta - \Lambda)$. It follows that $\Lambda = \Delta - \mho(\Theta')$ and (1) $\Theta' \subset \Theta$. By the right-left direction of Theorem 13, (2) Θ' is a choice set of $\Sigma(\Gamma \cup \Delta^{\bullet})$. By (1) and (2), $\Theta \notin \Phi(\Gamma \cup \Delta^{\bullet})$,—a contradiction.

($\Leftarrow$) Suppose $\Delta - \mho(\Theta) \in \mathbb{M}(\Gamma, \Delta)$. Hence $\Gamma \cup (\Delta - \mho(\Theta)) \nvdash \bot$. By the right-left direction of Theorem 13, Θ is a choice set of $\Sigma(\Gamma \cup \Delta^{\bullet})$. Assume that Θ is not a minimal choice set of $\Sigma(\Gamma \cup \Delta^{\bullet})$. Let $\Theta' \subset \Theta$ be a choice set of $\Sigma(\Gamma \cup \Delta^{\bullet})$. By the left-right direction of Theorem 13, $\Gamma \cup (\Delta - \mho(\Theta')) \nvdash \bot$. Note that $\Delta - \mho(\Theta) \subset \Delta - \mho(\Theta')$. Hence $\Delta - \mho(\Theta) \notin \mathbb{M}(\Gamma, \Delta)$—a contradiction. ■

We can now prove Theorem 5. Suppose $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C}$ and $\mathcal{A}$ are $\mathbf{CL}$-consistent sets. By Theorem 11 (where $\mathcal{G} = \emptyset$), $\mathcal{A}^{\mathsf{in}} \cup \mathcal{C}^{\mathsf{con}} \nvdash \bot$. Hence we can rely on Theorem 14, letting $\Gamma = \mathcal{A}^{\mathsf{in}} \cup \mathcal{C}^{\mathsf{con}}$ and $\Delta = \mathcal{G}^{\rightarrow}$, such that $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} = \Gamma \cup \Delta^{\bullet}$.

($\subseteq$) Suppose $\Theta \in \Phi(\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}})$. By the left-right direction of Theorem 14, $\mathcal{H}^{\rightarrow} = \mathcal{G}^{\rightarrow} - \mho(\Theta)$ is a maximal subset of $\mathcal{G}^{\rightarrow}$ for which $\mathcal{A}^{\mathsf{in}} \cup \mathcal{C}^{\mathsf{con}} \cup \mathcal{H}^{\rightarrow} \nvdash \bot$. In other words, $\mathcal{H}$ is a maximal subset of $\mathcal{G}$ such that $\mathcal{H}^{\rightarrow} \cup \mathcal{A}^{\mathsf{in}} \cup \mathcal{C}^{\mathsf{con}} \nvdash \bot$. By Theorem 11, $\mathcal{H}$ is a maximal subset of $\mathcal{G}$ such that $\mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A}) \cup \mathcal{C} \nvdash_{\mathbf{CL}} \bot$. By Definition 5, $\mathcal{H} \in \mathsf{maxfamily}(\mathcal{G}, \mathcal{A}, \mathcal{C})$. By Lemma 4 (where $\Gamma = \mathcal{A}^{\mathsf{in}} \cup \mathcal{C}^{\mathsf{con}}$ and $\Delta = \mathcal{G}^{\rightarrow}$) we immediately get $\Theta \subseteq \Omega_{\bullet}(\mathcal{G})$ and hence $\Theta = \Omega_{\bullet}(\mathcal{G} - \mathcal{H})$.

($\supseteq$) Let $\mathcal{H} \in \mathsf{maxfamily}(\mathcal{G}, \mathcal{A}, \mathcal{C})$. By Definition 5, $\mathcal{H}$ is a maximal subset of $\mathcal{G}$ such that $\mathrm{out}_{\mathsf{R}}(\mathcal{G}, \mathcal{A}) \cup \mathcal{C} \nvdash_{\mathbf{CL}} \bot$. By Theorem 11, ($\star$) $\mathcal{H}^{\rightarrow}$ is a maximal subset of $\mathcal{G}^{\rightarrow}$ such that $\mathcal{H}^{\rightarrow} \cup \mathcal{A}^{\mathsf{in}} \cup \mathcal{C}^{\mathsf{con}} \nvdash \bot$. Let $\Theta = \Omega_{\bullet}(\mathcal{G} - \mathcal{H})$. Hence, $\mathcal{H}^{\rightarrow} = \mathcal{G}^{\rightarrow} - \mho(\Theta)$. By ($\star$) and the right-left direction of Theorem 14, $\Theta \in \Phi\big(\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}}\big)$.

Proof of Theorem 6

Lemma 5. *Where $\Gamma \subseteq \mathcal{W}^{\not\bullet}$, $\Delta \subseteq \mathcal{W}^{\rightarrow}$, and $\Delta' \subseteq \Delta$: $\Gamma \cup \Delta' \vdash A$ iff there is a $\Theta \subseteq_f \Omega_{\bullet}(\Delta')$ such that $\Gamma \cup \Delta^{\bullet} \vdash A \vee \bigvee \Theta$.*

Proof. ($\Rightarrow$) Suppose $\Gamma \cup \Delta' \vdash A$. By the compactness of $\mathbf{CMIO}_{\mathsf{R}}$, there is a $\Delta'' \subseteq_f \Delta'$ such that $\Gamma \cup \Delta'' \vdash A$. Let $\Theta = \Omega_\bullet(\Delta'')$, whence $\Theta \subseteq_f \Omega_\bullet(\Delta')$. By the deduction theorem, $\Gamma \vdash A \vee \bigvee\{\neg B \mid B \in \Delta''\}$. Note that for all $B \in \Delta''$, $\bullet B \in \Delta^\bullet$ since $\Delta'' \subseteq \Delta$. Hence $\Gamma \cup \Delta^\bullet \vdash A \vee \bigvee \Theta$.

($\Leftarrow$) Suppose $\Gamma \cup \Delta^\bullet \vdash A \vee \bigvee \Theta$, where $\Theta \subseteq_f \Omega_\bullet(\Delta')$. Hence, $\Gamma \cup \Delta^\bullet \vdash A \vee \bigvee\{\neg B \mid B \in \mho(\Theta)\}$. By Fact 11, $\Gamma \vdash A \vee \bigvee\{\neg B \mid B \in \mho(\Theta)\}$. Hence by the monotonicity of $\mathbf{CMIO}_{\mathsf{R}}$ and disjunctive syllogism, $\Gamma \cup \mho(\Theta) \vdash A$. Since $\mho(\Theta) \subseteq \Delta'$ and by the monotonicity of $\mathbf{CMIO}_{\mathsf{R}}$, $\Gamma \cup \Delta' \vdash A$. ■

Letting $\Gamma = \mathcal{A}^{\mathsf{in}} \cup \mathcal{C}^{\mathsf{con}}$, $\Delta = \mathcal{G}^{\rightarrow}$, $\Delta' = \mathcal{H}^{\rightarrow}$, we get by Lemma 5:

Corollary 3. *Where* $\mathcal{H} \subseteq \mathcal{G}$, $\Gamma^{\mathcal{H},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash \mathsf{out}A$ *iff there is a* $\Theta \subseteq_f \Omega_\bullet(\mathcal{H})$ *such that* $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash \mathsf{out}A \vee Dab(\Theta)$.

Theorem 6 follows directly by Theorem 12 and Corollary 3. To see this, suppose $\mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A}) \cup \mathcal{C}$ and $\mathcal{A}$ are **CL**-consistent sets where $\mathcal{H} \subseteq \mathcal{G}$. We have: $A \in \mathrm{out}_{\mathsf{R}}(\mathcal{H}, \mathcal{A})$ iff [by Theorem 12] $\Gamma^{\mathcal{H},\mathcal{A}} \cup \mathcal{C}^{\mathsf{con}} \vdash \mathsf{out}A$ iff [by Corollary 3] there is a $\Theta \subseteq_f \Omega_\bullet(\mathcal{H})$ such that $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash \mathsf{out}A \vee Dab(\Theta)$.

Appendix 4: Proof of Theorem 8

The following property is known to hold for adaptive logics with the reliability strategy.[46]

Theorem 15. $\Gamma \vdash_{\mathbf{MIO}^{\Cap}_{\mathsf{R}}} A$ *iff there is a* $\Delta \subseteq_f \Omega_\bullet - \bigcup \Phi(\Gamma)$ *such that* $\Gamma \vdash_{\mathbf{CMIO}_{\mathsf{R}}} A \vee Dab(\Delta)$.

By Lemma 2, we have:

Corollary 4. $\Gamma \vdash_{\mathbf{MIO}^{\Cap}_{\mathsf{R}}} A$ *iff there is a* $\Delta \subseteq_f \Omega_\bullet(\mathcal{G}) - \bigcup \Phi(\Gamma)$ *such that* $\Gamma \vdash_{\mathbf{CMIO}_{\mathsf{R}}} A \vee Dab(\Delta)$.

From Theorem 5, we can derive:

Theorem 16. *Where* $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C}$ *and* $\mathcal{A}$ *are* **CL**-*consistent sets or (F)* $\in \hat{\mathsf{R}}$:

$$\bigcup \Phi\left(\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}}\right) = \Omega_\bullet\left(\mathcal{G} - \bigcap \mathsf{maxfamily}(\mathcal{G}, \mathcal{A}, \mathcal{C})\right)$$

We now prove Theorem 8.

For special case in which there are no maxfamilies we can use Lemma 1. If $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C}$ is **CL**-inconsistent then $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash \bot$ and hence $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}^{\Cap}_{\mathsf{R}}} \bot$. Hence, Theorem 8 holds for the case in which $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C}$ is **CL**-inconsistent.

Suppose now that $\mathrm{out}_{\mathsf{R}}(\emptyset, \mathcal{A}) \cup \mathcal{C}$ and $\mathcal{A}$ are **CL**-consistent sets. The following four properties are equivalent by Corollary 4 (item 1 and 2), Theorem 16 (items 2 and 3), and Theorem 6 (items 3 and 4):

1. $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{MIO}^{\Cap}_{\mathsf{R}}} \mathsf{out}A$

[46] It follows immediately from Theorems 6 and 11.5 in [6].

2. There is a $\Delta \subseteq_f \Omega_\bullet(\mathcal{G}) - \bigcup \Phi(\Gamma)$ such that $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_\mathsf{R}} \mathsf{out}A \vee \mathit{Dab}(\Delta)$.
3. There is a $\Delta \subseteq_f \Omega_\bullet(\bigcap \mathsf{maxfamily}(\mathcal{G},\mathcal{A},\mathcal{C}))$ such that $\Gamma^{\mathcal{G},\mathcal{A},\mathcal{C}} \vdash_{\mathbf{CMIO}_\mathsf{R}} \mathsf{out}A \vee \mathit{Dab}(\Delta)$.
4. $A \in \mathrm{out}_\mathsf{R}(\bigcap \mathsf{maxfamily}(\mathcal{G},\mathcal{A},\mathcal{C}),\mathcal{A})$.

References

[1] Amgoud, L., and H. Prade, Towards a logic of argumentation, in E. Hüllermeier, S. Link, T. Fober, and B. Seeger (eds.), *Scalable Uncertainty Management*, vol. 7520 of *Lecture Notes in Computer Science*, Springer, Berlin, 2012, pp. 558–565.

[2] Apothéloz, D., P.-Y. Brandt, and G. Quiroz, The function of negation in argumentation, *Journal of Pragmatics* 19(1):23–38, 1993.

[3] Batens, D., A survey of inconsistency-adaptive logics, in D. Batens, G. Priest, and J.-P. van Bendegem (eds.), *Frontiers of Paraconsistent Logic*, Research Studies Press, Kings College Publication, Baldock, 2000, pp. 49–73.

[4] Batens, D., The need for adaptive logics in epistemology, in *Logic, Epistemology, and the Unity of Science*, Springer, Berlin, 2004, pp. 459–485.

[5] Batens, D., A procedural criterion for final derivability in inconsistency-adaptive logics, *Journal of Applied Logic* 3:221–250, 2005.

[6] Batens, D., A universal logic approach to adaptive logics, *Logica Universalis* 1:221–242, 2007.

[7] Batens, D., Towards a dialogic interpretation of dynamic proofs, in C. Dégremont, L. Keiff, and H. Rückert (eds.), *Dialogues, Logics and Other Strange Things. Essays in Honour of Shahid Rahman*, College Publications, London, 2009, pp. 27–51.

[8] Batens, D., and D. Provijn, Pushing the search paths in the proofs. A study in proof heuristics, *Logique et Analyse* 173–175:113–134, 2001.

[9] Beirlaen, M., and C. Straßer, Two adaptive logics of norm-propositions, *Journal of Applied Logic* 11(2):147–168, 2013.

[10] Beirlaen, M., and C. Straßer, Nonmonotonic reasoning with normative conflicts in multi-agent deontic logic, *Journal of Logic and Computation* 24(6):1179–1207, 2014.

[11] Beirlaen, M., C. Straßer, and J. Meheus, An inconsistency-adaptive deontic logic for normative conflicts, *Journal of Philosophical Logic* 42(2):285–315, 2013.

[12] Bochman, A., *Explanatory Nonmonotonic Reasoning*, OECD Publishing, Paris, 2005.

[13] Boella, G., and L. van der Torre, Institutions with a hierarchy of authorities in distributed dynamic environments, *Artifical Intelligence and Law* 16:53–71, 2008.

[14] Boutilier, C., Conditional logics of normality: a modal approach, *Artificial Intelligence* 68(1):87–154, 1994.

[15] Brewka, G., Preferred subtheories: an extended logical framework for default reasoning, in *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, 1989, pp. 1043–1048.

[16] Crocco, G., and P. Lamarre, On the connection between non-monotonic inference systems and conditional logics, in B. Nebel and E. Sandewall (eds.), *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, 1992, pp. 565–571.

[17] Gabbay, D., Conditional implications and non-monotonic consequence, in G. Crocco, L. Farinhas del Cerro and A. Herzig (eds.), *Conditionals: From Philosophy to Computer Science*, vol. 5 of *Studies in Logic and Computation*, Oxford University Press, Oxford, 1995, pp. 337–359.

[18] Goble, L., Deontic logic (adapted) for normative conflicts, *Logic Journal of IGPL* 22(2):206–235, 2014.

[19] Gonçalves, R., and J. J. Alferes, An embedding of input-output logic in deontic logic programs, in T. Ågotnes, J. Broersen, and D. Elgesem (eds.), *Proceedings of the 11th International Conference on Deontic Logic in Computer Science*, vol. 7393 of *Lecture Notes in Artificial Intelligence*, Springer, Berlin, 2012, pp. 61–75.

[20] Hansen, J., G. Pigozzi, and L. van der Torre, Ten philosophical problems in deontic logic, in G. Boella, L. van der Torre, and H. Verhagen (eds.), *Normative Multi-agent Systems, No. 07122 in Dagstuhl Seminar Proceedings*, Dagstuhl, Germany, 2007, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[21] Horty, J. F., Moral dilemmas and nonmonotonic logic, *Journal of Philosophical Logic* 23(1):35–65, 1994.

[22] Horty, J. F., Skepticism and floating conclusions, *Artificial Intelligence* 135:55–72, 2002.

[23] Horty, J. F., *Reasons as Defaults*, Oxford University Press, Oxford, 2012.

[24] Kraus, S., D. J. Lehmann, and M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics, *Artificial Intelligence* 44:167–207, 1990.

[25] Lehmann, D. J., and M. Magidor, What does a conditional knowledge base entail?, *Artificial Intelligence* 55(1):1–60, 1992.

[26] Lin, F., and Y. Shoham, *Epistemic Semantics for Fixed-Points Non-monotonic Logics*, Morgan Kaufmann Publishers Inc., Pacific Grove, CA, 1990.

[27] Lindström, S., A semantic approach to nonmonotonic reasoning: inference operations and choice, *Uppsala Prints and Reprints in Philosophy* 10, 1994.

[28] Makinson, D., and K. Schlechta, Floating conclusions and zombie paths: two deep difficulties in the "directly skeptical" approach to defeasible inheritance nets, *Artificial Intelligence* 48:199–209, 1991.

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

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

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

[32] Makinson, D., and L. van der Torre, What is input/output logic? input/output logic, constraints, permissions, in G. Boella, L. van der Torre, and H. Verhagen (eds.), *Normative Multi-agent Systems, No. 07122 in Dagstuhl Seminar Proceedings*, Dagstuhl, Germany, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

[33] Meheus, J., M. Beirlaen, and F. Van De Putte, Avoiding deontic explosion by contextually restricting aggregation, in G. Governatori and G. Sartor (eds.), *Proceedings of the 10th International Conference on Deontic Logic in Computer Science*, vol. 6181 of *Lecture Notes in Artificial Intelligence*, Springer, 2010, pp. 148–165.

[34] Meheus, J., C. Straßer, and P. Verdée, Which style of reasoning to choose in the face of conflicting information? *Journal of Logic and Computation*, 2013. doi:10.1093/logcom/ext030.

[35] Parent, X. , and L. Van Der Torre, Aggregative deontic detachment for normative reasoning, Short paper, to appear in *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*, 2014.

[36] Parent, X. , and L. Van Der Torre, Sing and dance!, in F. Cariani, D. Grossi, J. Meheus and X. Parent (eds.), *Deontic Logic and Normative Systems*. Springer, Berlin, 2014, pp. 149–165.

[37] Pollock, J., Defeasible reasoning, in J. E. Adler and L. J. Rips (eds.), *Reasoning. Studies of Human Inference and Its Foundations*, Cambridge University Press, Cambridge, 2008, pp. 451–470.

[38] Prakken, H., Intuitions and the modelling of defeasible reasoning: some case studies, in S. Benferhat and E. Giunchiglia (eds.), *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning*, 2002, pp. 91–102.

[39] Reiter, R., A logic for default reasoning, *Artificial Intelligence* 13:81–132, 1980.

[40] Rescher, N., and R. Manor, On inference from inconsistent premises, *Theory and Decision* 1:179–217, 1970.

[41] Stolpe, A., Normative consequence: the problem of keeping it whilst giving it up, in R. van der Meyden and L. van der Torre (eds.), *Proceedings of the 9th International Conference on Deontic Logic in Computer Science*, vol. 5076 of *Lecture Notes in Computer Science*, Springer, Berlin, 2008, pp. 174–188.

[42] Stolpe, A., *Norms and Norm-System Dynamics*, PhD Thesis, Department of Philosophy, University of Bergen, Norway, 2008.

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

[44] Straßer, C. A deontic logic framework allowing for factual detachment, *Journal of Applied Logic* 9(1):61–80, 2011.

[45] Straßer, C., *Adaptive Logic and Defeasible Reasoning. Applications in Argumentation, Normative Reasoning and Default Reasoning*, Springer, Berlin, 2014.

[46] Straßer, C., M. Beirlaen, and J. Meheus, Tolerating deontic conflicts by adaptively restricting inheritance, *Logique & Analyse* 219:477–506, 2012.

[47] Straßer, C., and F. Van De Putte, Proof theories for superpositions of adaptive logics, *Logique et Analyse* 25(230), 2015

[48] Van De Putte, F., Hierarchic adaptive logics, *Logic Journal of the IGPL* 20:45–72, 2012.

[49] Van De Putte, F., and C. Straßer, Extending the standard format of adaptive logics to the prioritized case, *Logique et Analyse* 55(220):601–641, 2012.

[50] Van De Putte, F., and C. Straßer, A logic for prioritized normative reasoning, *Journal of Logic and Computation* 23(3):563–583, 2013.

[51] Van De Putte, F., and P. Verdée. The dynamics of relevance: Adaptive belief revision, *Synthese* 187(1):1–42, 2012.

[52] Van Der Torre, L., and Y.-H. Tan, Cancelling and overshadowing: two types of defeasibility in defeasible deontic logic, in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, 1995, pp. 1525–1532.

[53] Van Fraassen, B. C., Values and the heart's command, *Journal of Philosophy* 70:5–19, 1973.

C. Straßer, M. Beirlaen, F. Van De Putte
Centre for Logic and Philosophy of Science
Ghent University
Ghent, Belgium
and
Institute for Philosophy II
Ruhr-University Bochum
Bochum, Germany
Christian.Strasser@ruhr-uni-bochum.de

M. Beirlaen
Instituto de Investigaciones Filosóficas
Universidad Nacional Autónoma de México
México, Mexico
mathieubeirlaen@gmail.com