

Rational Monotony in Input/Output Logic

Xu Li Liuwen Yu Leendert van der Torre^{1 2}

*University of Luxembourg
Esch-sur-Alzette, Luxembourg*

Abstract

Input/Output (I/O) logic is a well-known formalism for reasoning about conditional norms. One of the main challenges of normative reasoning is to accommodate Contrary-To-Duty (CTD) paradoxes. Constrained I/O logic is devised as a general mechanism for resolving conflicts in normative systems. In this paper, we propose an I/O logic intended only for CTD reasoning, and we show its advantages in some CTD scenarios compared to constrained I/O logic. The main idea is to replace the Strengthening of the Input (SI) rule in I/O logic with a form of rational monotony, which is called a Weaker Version of RM (wRM). Since the latter is a non-Horn rule, we define our output operations using the concept of “reduction” from answer set programming. Our main technical contribution is a representation result for a variant of so-called simple-minded I/O logic where SI is replaced by wRM.

Keywords: Input/output logic, rational monotony, answer set programming.

1 Introduction

Contrary-to-duty (CTD) paradoxes are one of main challenges of normative reasoning. CTD paradoxes involve obligations in circumstances where some primary obligations are violated. A typical CTD paradox is Forrester’s [2]: Smith ought not to kill Jones, but if Smith does kill Jones, then Smith ought to kill Jones gently. Suppose that Smith kills Jones. The problem is that, while intuitively consistent, Forrester’s paradox cannot be represented consistently in standard deontic logic.

Input/Output (I/O) logic [10,17] is a well-known formalism for reasoning about conditional norms. However, the four basic types of I/O operations proposed in Makinson and Van der Torre’s original paper [10] cannot represent consistently CTD paradoxes like Forrester’s. The reason is that they all satisfy the principle of Strengthening of the Input (SI). To deal with CTD paradoxes, Makinson and Van der Torre [11] introduced constrained I/O logic, where a

¹ {xu.li, liuwen.yu, leon.vandertorre}@uni.lu.

² We thank three reviewers for their comments. This work was supported by the Fonds National de la Recherche Luxembourg through the project Deontic Logic for Epistemic Rights (OPEN O20/14776480) and the LoDEx project (INTER/DFG/23/17415164/LODEX).

set of constraints is used to deal with CTD reasoning. The idea is to consider maximal subsets of the normative system that do not yield contradiction (with the constraints). This follows Horty's [7] approach of using default logic to generate multiple extensions to handle conflicts. It can be traced back to at least the work of Van Fraassen [20] in the early 1970s. As such, not only can constrained I/O logic handle CTD reasoning but it can also represent dilemmas and conflicts consistently.

In normative systems, it could be argued that CTD reasoning differs from the issue of norm conflict resolution (see Section 2 for details). A general mechanism to solve conflicts (like constrained I/O logic) may not be perfect for CTD reasoning in all scenarios, as we will see later. However, to our best knowledge, this problem has not been mentioned or discussed in the literature of I/O logic. In this paper, we restrict our attention to CTD reasoning only, and we address the question of how to develop a specific I/O logic framework for CTD reasoning. We then compare this framework with constrained I/O logic.

It is well known that to deal with CTD paradoxes, the SI rule must be weakened. In fact, this is a basic feature of Dyadic Standard Deontic Logic (DDL [6]), which is one of the early paradigms of CTD reasoning before the introduction of I/O logic. There are three logical systems for DDL with increasing deductive power, known as Åqvist's E/F/G systems [15,17]. The strongest system, G, features a weak form of SI, which is known as Rational Monotony (RM) [17]:

$$\neg \bigcirc (\neg \psi / \varphi) \rightarrow (\bigcirc(\chi / \varphi) \rightarrow \bigcirc(\chi / \varphi \wedge \psi)).$$

That is, if ψ is permitted in the context of φ , then whatever is obligatory in the context of φ is also obligatory in the context of $\varphi \wedge \psi$. The intuition is that since ψ is permitted in φ , the conjunction $\varphi \wedge \psi$ does not represent a CTD scenario. Therefore, SI should take effect in this situation.

Following DDL, this paper deals with CTD reasoning within the I/O logic framework by replacing the SI principle with (a form of) RM. The main technical challenge here is that RM is a non-Horn rule (because of the negative premise in RM), which makes it difficult to define a rational logical consequence relation. Lehmann and Magidor [9] noticed this problem, and their solution was to propose the notion of "rational closure". In this paper, we adopt a different approach. In particular, we employ the concept of "reduction" from answer set programming [4], one of the most successful paradigms of logical programming. Our main technical finding is that in simple-minded I/O logic (which is the simplest type of I/O logic introduced in [10]), replacing SI with RM always results in an unique answer set. We also provide the representation result for the unique answer set.

The paper is structured as follows. Section 2 discusses informally the difference between CTD obligations and norms describing exceptions. Section 3 introduces the necessary preliminaries on I/O logic. Section 4 introduces RM as an alternative to SI. Section 5 presents wRM, which is better aligned with CTD reasoning, and provides representation results for the corresponding out-

put operations. Section 6 compares our approach with constrained I/O logic and dyadic deontic logic. Section 7 concludes with a summary of this paper and future directions.

2 Contrary-to-Duty vs. Exceptions

As mentioned in the Introduction, CTD reasoning is different to reasoning about conflicts or dilemmas. In normative systems, dilemmas can happen for different reasons. Here, we consider dilemmas caused by norms that describe exceptions. To understand how they are different to CTD reasoning, consider the cottage example introduced by Prakken and Sergot [14]:

1. It is forbidden to have a fence around the cottage.
2. If there is a cliff, there should be a fence.
3. If there is a fence, it should be white.

If the cottage example above is modeled in a deontic logic that supports SI, the first two sentences create a dilemma when the cottage is situated next to a cliff. But because being next to a cliff is described as an exception to the first norm in the regulations, the intuitive conclusion we can draw here is that there should be a fence. The first and third sentences describe a typical CTD scenario, and there is no dilemma or conflict. What is described by the two sentences is merely the following: the ideal situation is no fence, the sub-ideal situation is a white fence, and the worst situation is another colored fence. For more in-depth discussions on the difference between norms that describe exceptions and CTD reasoning, the reader is referred to [19] and [16] (where they are called overridden and factual defeasibility respectively). The cottage example is also discussed at length in [18, Section 2.3].

This paper will focus only on CTD reasoning in the cottage example. In other words, the I/O logic system we propose will generate consistent outputs in the presence of sentences 1 and 3 (for the input that there is a fence), but not for 1 and 2 (for the input that there is a fence). This is different to constrained I/O logic which not only deals with CTD reasoning but also represents dilemmas or conflicts in a consistent way. However, constrained I/O logic may not be perfect for CTD reasoning in all scenarios, as suggested by the following variant of Chisholm's paradox [1]:

Example 2.1 Consider an agent, Jones, whose neighbors are ill.

- (1) It ought to be that Jones goes to the assistance of his neighbors;
- (2) It ought to be that Jones tells them he is coming;
- (3) If Jones does not go to the assistance of his neighbors, then he ought not to tell them he is coming.

If we model Example 2.1 in constrained I/O logic (see Example 6.1), it predicts that either Jones ought to tell and also ought not to tell, or that Jones has no obligation at all when he does not go to the assistance of his neighbors. However, our intuition tells us that Jones ought not to tell. As we shall see,

the framework in our paper generates the correct outputs for this example.

3 Preliminaries

We assume a finite non-empty set PROP of propositional variables or atoms. Let \mathcal{L} be the propositional language generated by PROP , and elements of \mathcal{L} are called formulas. For all sets of formulas $A \cup \{a\}$, $A \vdash a$ denotes that a is a logical consequence of A in classical propositional logic. $Cn(A) = \{a \mid A \vdash a\}$ is the set of all the logical consequences of A . Given two formulas a and b , $a \dashv b$ abbreviates that $a \vdash b$ and $b \vdash a$. We write $Eq(a) = \{b \mid a \dashv b\}$ for the set of all formulas that are equivalent to a . We write $b \prec a$ if: $a \vdash b$ but $b \not\vdash a$. Note that \prec is a well-founded relation³ on \mathcal{L} because we only have a finite number of propositional variables.

An output operation is a function $out : \wp(\mathcal{L} \times \mathcal{L}) \rightarrow \wp(\mathcal{L} \times \mathcal{L})$. Intuitively, a set $N \subseteq \mathcal{L} \times \mathcal{L}$ represents a normative system and the elements $(a, x) \in N$ is read as ‘‘given a , it ought to be the case that x ’’. The output $out(N)$ is intended to characterize the set of obligations that can be derived from N . We put $out(N, a) = \{x \mid (a, x) \in out(N)\}$ for all formulas a . Thus, $out(N, a)$ is the collection of (unconditional) obligations held in context a . Below, we list some well-known properties of out as described in the I/O logic literature:

- REF If $(a, x) \in N$, then $(a, x) \in out(N)$.
- T $(\top, \top) \in out(N)$, where \top abbreviates $p \vee \neg p$ for certain atom p .
- SI If $(a, x) \in out(N)$ and $b \vdash a$, then $(b, x) \in out(N)$.
- WO If $(a, x) \in out(N)$ and $x \vdash y$, then $(a, y) \in out(N)$.
- AND If $(a, x) \in out(N)$ and $(a, y) \in out(N)$, then $(a, x \wedge y) \in out(N)$.
- OR If $(a, x) \in out(N)$ and $(b, x) \in out(N)$, then $(a \vee b, x) \in out(N)$.
- CT If $(a, x) \in out(N)$ and $(a \wedge x, y) \in out(N)$, then $(a, y) \in out(N)$.

The four basic types of output operations, $out_1 - out_4$ [10], are defined as follows:

- $out_1(N)$ is the smallest set closed under {REF, T, SI, WO, AND};
- $out_2(N)$ is the smallest set closed under {REF, T, SI, WO, AND, OR};
- $out_3(N)$ is the smallest set closed under {REF, T, SI, WO, AND, CT};
- $out_4(N)$ is the smallest set closed under {REF, T, SI, WO, AND, OR, CT}.

Forrester’s paradox, introduced at the beginning of this paper, can be represented by the set $N = \{(\top, \neg k), (k, k \wedge g)\}$. For each $i \in \{1, 2, 3, 4\}$, we have $out_i(N, k) = Cn(\perp)$ because the pair $(k, \neg k)$ can be derived using SI. Thus, the four output operations are not suitable for CTD reasoning. In this paper, we aim to develop variants of the above output operations without the SI property. But simply removing SI from the above definition would not work because we would also lose other appealing properties like:

³ A binary relation \prec on a set X is well-founded if every non-empty subset $S \subseteq X$ has a minimal element with respect to \prec , i.e., there is $m \in S$ such that $s \not\prec m$ for all $s \in S$.

$$\begin{array}{ll} \text{IEQ} & \text{If } (a, x) \in \text{out}(N) \text{ and } a \dashv b, \text{ then } (b, x) \in \text{out}(N) \\ \text{AT} & (a, \top) \in \text{out}(N) \end{array}$$

Therefore, the following output operations $\text{out}_1^- - \text{out}_4^-$ will be our starting point.

Definition 3.1 For each $i \in \{1, 2, 3, 4\}$, let out_i^- be the output operation obtained by substituting AT with T and IEQ with SI in the definition of out_i . For example, $\text{out}_2^-(N)$ is the smallest set closed under {REF, AT, IEQ, WO, AND, OR}.

Note that by adding SI into out_i^- , we regain the output operation out_i . Below, we state the representation results for $\text{out}_1^- - \text{out}_3^-$. For every set $N \subseteq \mathcal{L} \times \mathcal{L}$ and a set A of formulas, let $N(A) = \{x \mid (a, x) \in N \text{ for some } a \in A\}$. Given a formula a , a finite non-empty set B of formulas is a *partition* of a if $a \dashv \bigvee B$. Let $\text{PAR}(a)$ be the set of all the partitions of a .

Proposition 3.2 *The following hold:*

- (1) $\text{out}_1^-(N, a) = \text{Cn}(N(\text{Eq}(a)))$
- (2) $\text{out}_2^-(N, a) = \text{Cn}\left(\bigcup_{B \in \text{PAR}(a)} \bigcap_{b \in B} \text{Cn}(N(b))\right)$
- (3) $\text{out}_3^-(N, a) = \bigcup_{i \in \mathbb{N}} A_i$ where
 - $A_0 = \text{out}_1^-(N, a)$
 - $A_{i+1} = \text{Cn}(A_i \cup \bigcup_{x \in A_i} \text{out}_1^-(N, a \wedge x))$

Proof. We show only (2). The proofs for (1) and (3) can be found in Appendix A. Let out be the output operation such that for all formulas a , $\text{out}(N, a) = \text{Cn}\left(\bigcup_{B \in \text{PAR}(a)} \bigcap_{b \in B} \text{Cn}(N(b))\right)$. We first show that $\text{out}(N)$ is closed under {REF, AT, IEQ, WO, AND, OR}. The only interesting case is OR. Suppose $x \in \text{out}(N, a)$ and $x \in \text{out}(N, b)$. According to the former, there are $B_1, \dots, B_m \in \text{PAR}(a)$ and $b_i \in \bigcap_{b \in B_i} \text{Cn}(N(b))$ for each B_i such that $b_1 \wedge \dots \wedge b_m \vdash x$. Similarly, by the latter, there are $C_1, \dots, C_n \in \text{PAR}(b)$ and $c_j \in \bigcap_{c \in C_j} \text{Cn}(N(c))$ for each C_j such that $c_1 \wedge \dots \wedge c_n \vdash x$. From propositional logic, it follows that

$$\bigwedge_{1 \leq i \leq m, 1 \leq j \leq n} b_i \vee c_j \vdash x \quad (1)$$

Note that for each $1 \leq i \leq m$ and $1 \leq j \leq n$, it is the case that $B_i \cup C_j \in \text{PAR}(a \vee b)$. Since $b_i \in \bigcap_{b \in B_i} \text{Cn}(N(b))$ and $c_j \in \bigcap_{c \in C_j} \text{Cn}(N(c))$, then $b_i \vee c_j \in \bigcap_{d \in B_i \cup C_j} \text{Cn}(N(d))$. Therefore, $x \in \text{out}(N, a \vee b)$ by the definition of out and (1).

It remains to demonstrate that $\text{out}(N)$ is the smallest. Let $\text{out}'(N)$ be a set closed under {REF, AT, IEQ, WO, AND, OR}. We show that $\text{out}(N, a) \subseteq$

$out'(N, a)$. Since $out'(N)$ is closed under AND and WO, it suffices to show that $\bigcap_{b \in B} Cn(N(b)) \subseteq out'(N, a)$ for each $B \in \text{PAR}(a)$. Note that, since $out'(N)$ is closed under REF, AND, and WO, then $out'(N, b) \supseteq Cn(N(b))$. Therefore, if $x \in \bigcap_{b \in B} Cn(N(b))$, then $x \in out'(N, b)$ for each $b \in B$. Since $out'(N)$ is closed under OR, then $x \in out'(N, \bigvee B)$. Note that $\bigvee B \dashv\vdash a$. Hence, by IEQ, $x \in out'(N, a)$. \square

Let us see how the above results can be applied to CTD paradoxes like Forrester's [2] and Chisholm's [1].

Example 3.3 [Forrester's paradox] Let $N = \{(\top, \neg k), (k, k \wedge g)\}$. Then

- $out_1^-(N, a) = Cn(\neg k)$ if $a \dashv\vdash \top$;
- $out_1^-(N, a) = Cn(k \wedge g)$ if $a \dashv\vdash k$;
- $out_1^-(N, a) = Cn(\emptyset)$ if $a \not\dashv\vdash \top$ and $a \not\dashv\vdash k$.

Example 3.4 [Chisholm's paradox] Let $N = \{(\top, g), (g, t), (\neg g, \neg t)\}$. Then

- $out_3^-(N, a) = Cn(g \wedge t)$ if $a \dashv\vdash \top$;
- $out_3^-(N, a) = Cn(t)$ if $a \dashv\vdash g$;
- $out_3^-(N, a) = Cn(\neg t)$ if $a \dashv\vdash \neg g$;
- $out_3^-(N, a) = Cn(\emptyset)$ if $a \not\dashv\vdash \top$, $a \not\dashv\vdash g$, and $a \not\dashv\vdash \neg g$;

Although the two CTD paradoxes can be consistently represented, the problem is that some meaningful conclusions are not derived. For example, in Forrester's paradox, if c is a proposition that is different to k and g (like "it is cloudy"), then intuitively, we still have the obligation not to kill given c . However, $\neg k \notin out_1^-(N, c)$ as $c \not\dashv\vdash \top$. The source of the problem is that we completely drop SI in $out_1^- - out_4^-$. As argued in [13], simply dropping SI is too heavy-handed. "We need to know why SI is not always appropriate, especially when it remains justified".

4 Rational Monotony

In this section, we consider how to add RM as an inference rule into out_1^- , and we show that RM is not suitable for certain CTD paradoxes like Forrester's.

To interpret RM in the context of I/O logic, we need to be clear about what we mean by permission. In the I/O logic framework, there are multiple notions of permission [12]. In this paper, we will mainly consider the notion of negative permission, which is defined in [12] as follows:

$$(a, x) \in negperm(N) \text{ iff } (a, \neg x) \notin out(N).$$

If we interpret the notion of permission in RM as negative permission, then RM can be interpreted as follows:

$$\text{RM} \quad \text{If } (a, \neg b) \notin out(N) \text{ and } (a, x) \in out(N), \text{ then } (a \wedge b, x) \in out(N).$$

We want to define the extension of out_1^- with RM. As before, we may define an output operation out_1^{rm} such that $out_1^{rm}(N)$ is the smallest set closed under

$\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}\}$. However, the following example shows that $\text{out}_1^{\text{rm}}(N)$ thus defined does not exist for certain N . This suggests that RM cannot be used in this way.

Example 4.1 Assume that we only have one propositional variable c in the language. Let $N = \{(\top, c)\}$. We show that $\text{out}_1^{\text{rm}}(N)$ as defined above does not exist. Consider the output operations out and out' defined as follows:

$$\text{out}(N, x) = \begin{cases} Cn(c) & \text{if } x \dashv\vdash \top \text{ or } x \dashv\vdash c; \\ Cn(\emptyset) & \text{otherwise.} \end{cases} \quad \text{out}'(N, x) = \begin{cases} Cn(\perp) & \text{if } x \dashv\vdash \top; \\ Cn(\emptyset) & \text{otherwise.} \end{cases}$$

We will show that $\text{out}_1^{\text{rm}}(N) = \text{out}(N) \cap \text{out}'(N)$ but that $\text{out}(N) \cap \text{out}'(N)$ is not closed under RM (which contradicts the definition of $\text{out}_1^{\text{rm}}(N)$).

Note that $\text{out}(N) \cap \text{out}'(N) = \text{out}_1^-(N)$ by Proposition 3.2. Therefore, $\text{out}_1^{\text{rm}}(N) \supseteq \text{out}(N) \cap \text{out}'(N)$. To show the inclusion from right to left, all we have to do is check that both $\text{out}(N)$ and $\text{out}'(N)$ are closed under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}\}$. Here, we only show the case for RM. We first show that $\text{out}(N)$ is closed under RM. Then suppose, toward a contradiction, that there are a, b, x such that $(a, \neg b) \notin \text{out}(N)$, $(a, x) \in \text{out}(N)$, and $(a \wedge b, x) \notin \text{out}(N)$. Then $\text{out}(N, a) \not\subseteq \text{out}(N, a \wedge b)$. By the definition of $\text{out}(N)$, it must be the case that $\text{out}(N, a) = Cn(c)$ and $\text{out}(N, a \wedge b) = Cn(\emptyset)$. So,

- (1) $a \dashv\vdash \top$ or $a \dashv\vdash c$;
- (2) $a \wedge b \not\dashv\vdash \top$ and $a \wedge b \not\dashv\vdash c$;
- (3) $\neg b \notin Cn(c)$.

By (3), $b \dashv\vdash c$ or $b \dashv\vdash \top$. In the former case, $a \wedge b \dashv\vdash c$, contradicting (2). In the latter case, if $a \dashv\vdash \top$, then $a \wedge b \dashv\vdash \top$, contradicting (2). Otherwise, $a \dashv\vdash c$, thus $a \wedge b \dashv\vdash c$, contradicting (2). Therefore, there cannot be such a, b, x . This shows that $\text{out}(N)$ is closed under RM. Similarly, we can show that $\text{out}'(N)$ is also closed under RM.

Therefore, it must be $\text{out}_1^{\text{rm}}(N) = \text{out}(N) \cap \text{out}'(N)$. However, $\text{out}(N) \cap \text{out}'(N)$ is not closed under RM because we have $\text{out}(N, \top) \cap \text{out}'(N, \top) = Cn(c)$ and $\text{out}(N, c) \cap \text{out}'(N, c) = Cn(\emptyset)$. But $(\top, \neg c) \notin \text{out}(N) \cap \text{out}'(N)$.

So, we may instead define $\text{out}_1^{\text{rm}}(N)$ as minimal sets closed under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}\}$. In Example 4.1, it can be seen that both $\text{out}(N)$ and $\text{out}'(N)$ are minimal sets closed under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}\}$. However, we have reasons for preferring out . In the context of \top , it is evident that the only obligation is c . In the context of c , since c is not forbidden in the context of \top , obligation c is inherited from the context of \top by the RM principle. Moreover, in the context of $\neg c$ and \perp , since both of them are prohibited in the context of \top , no obligation can be inherited from the context of \top .

To capture this intuition, it turns out that we can employ the same ideas from research on semantics for logic programming. We can regard the (a, x) pairs as atoms and the properties of the output operation (e.g., REF) as rules in logic programs. The definitions below follow the stable semantics for logic

programming [3].

Definition 4.2 Given a set $M \subseteq \mathcal{L} \times \mathcal{L}$, the *reduction* of RM to M is the following property $\text{RM}|_M$:

$$\text{RM}|_M \quad \text{If } (a, \neg b) \notin M \text{ and } (a, x) \in \text{out}(N), \text{ then } (a \wedge b, x) \in \text{out}(N).$$

Definition 4.3 Let $N \subseteq \mathcal{L} \times \mathcal{L}$. For all $M \subseteq \mathcal{L} \times \mathcal{L}$, let $\text{out}_1^M(N)$ be the smallest set closed under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}|_M\}$. If $M = \text{out}_1^M(N)$, we say M is a stable set of N under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}\}$.

Next, we illustrate the definition of “stable set” by Forrester’s paradox. In what follows, given a set $M \subseteq \mathcal{L} \times \mathcal{L}$ and a formula a , let $M(a) = \{x \mid (a, x) \in M\}$.

Example 4.4 [Forrester’s paradox] Let $N = \{(\top, \neg k), (k, k \wedge g)\}$. Let M be a set of pairs of formulas such that:

- $M(a) = Cn(\neg k)$ if $a \not\vdash k$,
- $M(a) = Cn(\perp)$ if $a \dashv k$.

We show that M is a stable set of N under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}\}$. We first show that M is closed under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}|_M\}$. The only interesting case is $\text{RM}|_M$. Suppose, toward a contradiction, that there exist a and b such that $(a, \neg b) \notin M$ and $M(a) \not\subseteq M(a \wedge b)$. By the former, $a \not\vdash k$. By the latter, $a \dashv k$. Contradiction!

But we also need to show that M is the smallest. Let M' be a set closed under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}|_M\}$. We show that $M(a) \subseteq M'(a)$ for all formulas a by considering the following cases:

- (1) $a \dashv \top$. It is trivial that $M(a) \subseteq M'(\top)$.
- (2) $a \not\vdash \top$. We consider the following sub-cases:
 - (a) $a \not\vdash k$. Then $\neg a \notin Cn(\neg k) = M(\top)$. Since M' is closed under $\text{RM}|_M$, then $M'(\top) \subseteq M'(\top \wedge a) = M'(a)$. Thus, $M(a) = M(\top) \subseteq M'(\top) \subseteq M'(a)$.
 - (b) $a \vdash k$ and $k \not\vdash a$. Note that $a \dashv (a \vee g) \wedge (a \vee \neg g)$. Therefore, $M'(a) = M'((a \vee g) \wedge (a \vee \neg g))$ by IEQ. Since $a \vee g \not\vdash k$, $M(a \vee g) = Cn(\neg k)$, then $\neg(a \vee \neg g) \notin M(a \vee g)$. By $\text{RM}|_M$, $M'(a \vee g) \subseteq M'((a \vee g) \wedge (a \vee \neg g)) = M'(a)$. Note that $a \vee g \not\vdash \top$ and $a \vee g \not\vdash k$, and by (a) we have $M(a \vee g) \subseteq M'(a \vee g)$. Therefore, $M(a) = M(a \vee g) \subseteq M'(a \vee g) \subseteq M'(a)$.
 - (c) $a \dashv k$. By REF and WO, $k \in M'(k)$. On the other hand, since $\neg(k \vee \neg g) \notin M(k \vee g)$, by $\text{RM}|_M$ it follows that $M'(k \vee g) \subseteq M'((k \vee g) \wedge (k \vee \neg g)) = M'(k)$. Since $Cn(\neg k) = M(k \vee g) \subseteq M'(k \vee g)$ by (a), then $\neg k \in M'(k)$. Thus, by AND and WO, $M'(k) = Cn(\perp) = M(a)$.

The above example suggests that RM is too strong to represent Forrester’s paradoxes because it predicts that we ought to both kill and not kill in the context of k . RM does not prevent the derivation of the pair $(k, \neg k)$. This can be explained by Figure 1. On the left, we see that the direct derivation

of $(k, \neg k)$ from $(\top, \neg k)$ is blocked because k is forbidden in the context of \top . However, $(k, \neg k)$ can still be derived from $(\top, \neg k)$ via an intermediate step $(k \vee g, \neg k)$, as shown to the right of Figure 1. The crucial step here is from $(k \vee g, \neg k)$ to $((k \vee g) \wedge (k \vee \neg g), \neg k)$. The reason is that even if the formula $(k \vee g) \wedge (k \vee \neg g)$ itself is forbidden in the context of $k \vee g$, their difference $k \vee \neg g$ is still permitted. Thus, the pair $((k \vee g) \wedge (k \vee \neg g), \neg k)$ (which, by IEQ, is equivalent to $(k, \neg k)$) can still be derived.

Remark 4.5 The above argument is specific to set M . One may wonder whether there exist other stable sets M' of N under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}\}$ such that the pair $(k, \neg k)$ does not belong to M' . The answer is No. We can show that M is the unique stable set. The reason will become clear in Section 5.

$$\frac{\frac{\frac{(\top, \neg k)}{(k, \neg k)}}{\frac{\frac{(\top, \neg k)}{(k \vee g, \neg k)} \text{RM}|_M}{\frac{\frac{((k \vee g) \wedge (k \vee \neg g), \neg k)}{(k, \neg k)} \text{RM}|_M}{\text{IEQ}}}}{}}$$

Fig. 1. RM allows the derivation of $(k, \neg k)$.

5 A Weaker Version of RM

To deal with CTD scenarios, we consider the following weaker version of RM, which we call wRM. To obtain wRM, we just replace the premise $(a, \neg b) \notin \text{out}(N)$ in RM with $(a, \neg(a \wedge b)) \notin \text{out}(N)$, thereby ensuring that $a \wedge b$ as a whole is permitted in the general context of a .

wRM If $(a, \neg(a \wedge b)) \notin \text{out}(N)$ and $(a, x) \in \text{out}(N)$,
then $(a \wedge b, x) \in \text{out}(N)$.

Proposition 5.1 *The following hold:*

- (i) *Let $\text{out}(N)$ be closed under WO. If $\text{out}(N)$ is closed under RM, then it is closed under wRM.*
- (ii) *Let $\text{out}(N)$ be closed under WO, AND and ID (given below). Then $\text{out}(N)$ is closed under RM iff it is closed under wRM.*

ID $(a, a) \in \text{out}(N)$ for all formulas a .

Proof. We give the proof for (ii) only. The implication from left to right follows from (i). From right to left: we just need to show that $(a, \neg b) \notin \text{out}(N)$ implies $(a, \neg(a \wedge b)) \notin \text{out}(N)$. We show the contrapositive, i.e., $(a, \neg(a \wedge b)) \in \text{out}(N)$ implies $(a, \neg b) \in \text{out}(N)$. Suppose $(a, \neg(a \wedge b)) \in \text{out}(N)$. Since $\text{out}(N)$ is closed under ID and AND, $(a, a \wedge \neg(a \wedge b)) \in \text{out}(N)$. Since $a \wedge \neg(a \wedge b) \vdash \neg b$, then $(a, \neg b) \in \text{out}(N)$ by WO. \square

Analogously to Definition 4.2, we can define the ‘‘reduction’’ of wRM with respect to a set of pairs of formulas as follows:

Definition 5.2 Given a set $M \subseteq \mathcal{L} \times \mathcal{L}$, the *reduction* of wRM to M is the following property $wRM|_M$:

$$wRM|_M \quad \text{If } (a, \neg(a \wedge b)) \notin M \text{ and } (a, x) \in out(N), \\ \text{then } (a \wedge b, x) \in out(N).$$

Thus, the notions of a “stable set” can be defined as follows:

Definition 5.3 Let $N \subseteq \mathcal{L} \times \mathcal{L}$ and $\mathbb{P} \subseteq \{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{OR}, \text{CT}\}$. For all sets $M \subseteq \mathcal{L} \times \mathcal{L}$, let $out^M(N)$ be the smallest set closed under $\mathbb{P} \cup \{wRM|_M\}$. If $M = out^M(N)$, we say M is a stable set of N under $\mathbb{P} \cup \{wRM\}$.

In what follows, we will focus on stable sets under the following four sets of properties $\mathbb{P}_1 - \mathbb{P}_4$:

- $\mathbb{P}_1 = \{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}\}$,
- $\mathbb{P}_2 = \{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{OR}\}$,
- $\mathbb{P}_3 = \{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{CT}\}$,
- $\mathbb{P}_4 = \{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{OR}, \text{CT}\}$.

5.1 The Case of \mathbb{P}_1

In this subsection, we show that for all normative systems N , there is always an unique stable set of N under $\mathbb{P}_1 \cup \{wRM\}$. Furthermore, we can give a representation result for the stable set.

Definition 5.4 Let a set $N \subseteq \mathcal{L} \times \mathcal{L}$ be given. We define an output operation $out_1^{wr}(N)$ inductively as follows:

- If $a \dashv\vdash \top$, then $out_1^{wr}(N, a) = Cn(N(Eq(\top)))$;
- $out_1^{wr}(N, a) = Cn \left(N(Eq(a)) \cup \bigcup_{\{b | b \prec a \& \neg a \notin out_1^{wr}(N, b)\}} out_1^{wr}(N, b) \right)$.

Example 5.5 [Forrester’s paradox] Let $N = \{(\top, \neg k), (k, k \wedge g)\}$. For simplicity, we assume that there are only two propositional variables k and g in the language. Then we have:

- $out_1^{wr}(N, \top) = Cn(\neg k)$;
- $out_1^{wr}(N, \neg(\neg k \wedge \neg g)) = Cn(\neg k)$;
- $out_1^{wr}(N, \neg(\neg k \wedge g)) = Cn(\neg k)$;
- $out_1^{wr}(N, k) = Cn(k \wedge g)$.

Furthermore, $out_1^{wr}(N)$ is completely given as follows:

- $out_1^{wr}(N, a) = Cn(\neg k)$ if $a \not\vdash k$;
- $out_1^{wr}(N, a) = Cn(k \wedge g)$ if $a \vdash k$ and $a \not\vdash k \wedge \neg g$;
- $out_1^{wr}(N, a) = Cn(\emptyset)$ if $a \vdash k \wedge \neg g$.

Next, we show that $out_1^{wr}(N)$ is the unique stable set of N under $\mathbb{P}_1 \cup \{wRM\}$.

Lemma 5.6 $out_1^{wr}(N)$ is a stable set of N under $\mathbb{P}_1 \cup \{wRM\}$.

Proof. We need to demonstrate that $out_1^{wr}(N)$ is the smallest set closed under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}|_{out_1^{wr}(N)}\}$. We first show that $out_1^{wr}(N)$ is closed under the set of properties. The only interesting case is $\text{RM}|_{out_1^{wr}(N)}$. Suppose $(a, \neg(a \wedge b)) \notin out_1^{wr}(N)$. We need to show that $out_1^{wr}(N, a) \subseteq out_1^{wr}(N, a \wedge b)$. (1) If $a \dashv a \wedge b$, then $out_1^{wr}(N, a) = out_1^{wr}(N, a \wedge b)$ since $out_1^{wr}(N)$ is closed under IEQ. (2) Otherwise, we have $a \prec a \wedge b$. Since $(a, \neg(a \wedge b)) \notin out_1^{wr}(N)$, by the definition of $out_1^{wr}(N, a \wedge b)$, it follows that $out_1^{wr}(N, a) \subseteq out_1^{wr}(N, a \wedge b)$.

We then have to demonstrate that $out_1^{wr}(N)$ is the smallest. Let $out(N)$ be an arbitrary set closed under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{RM}|_{out_1^{wr}(N)}\}$. We prove that $out_1^{wr}(N, a) \subseteq out(N, a)$ for all formulas a , by induction on the relation \prec . The base case where $a \dashv \top$ is trivial. Suppose that for all $b \prec a$, $out_1^{wr}(N, b) \subseteq out(N, b)$. We are going to show that $out_1^{wr}(N, a) \subseteq out(N, a)$. Since $out(N)$ is closed under AND and WO, we just have to show the following:

$$N(Eq(a)) \cup \bigcup_{\{b|b \prec a \& \neg a \notin out_1^{wr}(N, b)\}} out_1^{wr}(N, b) \subseteq out(N, a)$$

It is obvious that $N(Eq(a)) \subseteq out(N, a)$ since $out(N)$ is closed under REF and IEQ. Let $b \prec a$ and $\neg a \notin out_1^{wr}(N, b)$. Since $\neg(b \wedge a) \notin out_1^{wr}(N, b)$ by WO, $out(N, b) \subseteq out(N, b \wedge a)$ as $out(N)$ is closed under $\text{RM}|_{out_1^{wr}(N)}$. Thus, by IEQ, $out(N, b) \subseteq out(N, a)$. Note that $out_1^{wr}(N, b) \subseteq out(N, b)$ by IH. Therefore, $out_1^{wr}(N, b) \subseteq out(N, a)$. \square

Theorem 5.7 For all sets $N, M \subseteq \mathcal{L} \times \mathcal{L}$, M is a stable set of N under $\mathbb{P}_1 \cup \{\text{wRM}\}$ iff $M = out_1^{wr}(N)$.

Proof. The direction from right to left follows from Lemma 5.6. From left to right. Let M be a stable set of N under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{wRM}\}$. We first define an auxiliary output operation $out(N)$ as follows:

- if $a \dashv \top$, then $out(N, a) = Cn(N(Eq(\top)))$
- $out(N, a) = Cn \left(N(Eq(a)) \cup \bigcup_{\{b|b \prec a \& \neg a \notin M(b)\}} out(N, b) \right)$

It can be verified that $out(N)$ is the smallest set closed under $\{\text{REF}, \text{AT}, \text{IEQ}, \text{WO}, \text{AND}, \text{wRM}|_M\}$. Thus, $M = out(N)$. By induction on the relation \prec , it can be shown that $out(N, a) = out_1^{wr}(N, a)$ for all formulas a . Therefore, $M = out_1^{wr}(N)$. \square

Remark 5.8 The result in Theorem 5.7 can be extended to cases where wRM is replaced by RM. That is, M is a stable set of N under $\mathbb{P}_1 \cup \{\text{RM}\}$ iff $M = out_1^r(N)$, where out_1^r is given as follows:

- if $a \dashv \top$, then $out_1^r(N, a) = Cn(N(Eq(\top)))$,
- $out_1^r(N, a) = Cn(N(Eq(a)) \cup \bigcup_{\{b|b \prec a \& b \wedge \neg a \notin out_1^r(N, b)\}} out_1^r(N, b))$.

This answers the question raised in Remark 4.5. We omitted the proof because it is similar to that of Theorem 5.7.

5.2 The Cases of $\mathbb{P}_2 - \mathbb{P}_4$

Given Theorem 5.7, one may wonder if similar results hold also in the cases of $\mathbb{P}_2 - \mathbb{P}_4$. Formally, we could define the output operations out_2^{wr} , out_3^{wr} , and out_4^{wr} analogously to Definition 5.4.

Definition 5.9 Let a set $N \subseteq \mathcal{L} \times \mathcal{L}$ be given. For each $i \in \{2, 3, 4\}$, we define $out_i^{wr}(N)$ inductively as follows:

- if $a \dashv\vdash \top$, then $out_i^{wr}(N, a) = out_i^-(N, \top)$;
- $out_i^{wr}(N, a) = Cn \left(out_i^-(N, a) \cup \bigcup_{\{b \mid b \prec a \& \neg a \notin out_i^{wr}(N, b)\}} out_i^{wr}(N, b) \right)$.

Unfortunately, in general, $out_i^{wr}(N)$ thus defined may not result in a stable set of N under the corresponding set of properties, as shown in the next example.

Example 5.10 Assume that there are only two propositional variables a, x in the language. Let $N = \{(\top, x), (a \wedge x, a)\}$. We show that $out_3^{wr}(N)$ is not closed under CT. We have:

- $x \in out_3^{wr}(N, a)$;
- $a \in out_3^{wr}(N, a \wedge x)$;
- $a \notin out_3^{wr}(N, a) = Cn(x)$.

Similar examples of N can be proposed to show that $out_2^{wr}(N)$ may not be closed under WO and likewise for $out_4^{wr}(N)$.

The next proposition states sufficient conditions for $out_i^{wr}(N)$ to be a stable set of N under the corresponding set of properties.

Proposition 5.11 For any set $N \subseteq \mathcal{L} \times \mathcal{L}$, the following hold:

- (1) if $out_2^{wr}(N)$ is closed under OR, then $out_2^{wr}(N)$ is a stable set of N under $\mathbb{P}_2 \cup \{\text{wRM}\}$.
- (2) if $out_3^{wr}(N)$ is closed under CT, then $out_3^{wr}(N)$ is a stable set of N under $\mathbb{P}_3 \cup \{\text{wRM}\}$.
- (3) if $out_4^{wr}(N)$ is closed under OR and CT, then $out_4^{wr}(N)$ is a stable set of N under $\mathbb{P}_4 \cup \{\text{wRM}\}$.

Proof. The proofs are similar to that of Lemma 5.6. \square

The above result can be applied to Chisholm's paradox.

Example 5.12 [Chisholm's paradox] Let $N = \{(\top, g), (g, t), (\neg g, \neg t)\}$. Then we have the following (the proof can be found in Appendix B):

- $out_3^{wr}(N, a) = Cn(g \wedge t)$ if $a \not\vdash \neg g \vee \neg t$.
- $out_3^{wr}(N, a) = Cn(\emptyset)$ if $a \vdash \neg g \vee \neg t$ and $a \not\vdash \neg g$.
- $out_3^{wr}(N, a) = Cn(\neg t)$ if $a \vdash \neg g$ and $a \not\vdash \neg g \wedge t$.
- $out_3^{wr}(N, a) = Cn(\emptyset)$ if $a \vdash \neg g \wedge t$.

To demonstrate that $out_3^{wr}(N)$ is a stable set of N under $\mathbb{P}_3 \cup \{\text{wRM}\}$, by Proposition 5.11 all we need to show is that $out_3^{wr}(N)$ is closed under CT. Let $x \in out_3^{wr}(N, a)$, we just need to show that $out_3^{wr}(N, a \wedge x) \subseteq out_3^{wr}(N, a)$. We consider the following cases:

- (1) $out_3^{wr}(N, a) = Cn(\emptyset)$. Then $a \wedge x \dashv\vdash a$. Therefore, $out_3^{wr}(N, a \wedge x) = out_3^{wr}(N, a)$.
- (2) $out_3^{wr}(N, a) = Cn(g \wedge t)$. Then $a \not\vdash \neg g \vee \neg t$. Since $g \wedge t \vdash x$, then $a \wedge x \not\vdash \neg g \vee \neg t$. Thus, $out_3^{wr}(N, a \wedge x) = Cn(g \wedge t) = out_3^{wr}(N, a)$.
- (3) $out_3^{wr}(N, a) = Cn(\neg t)$. Then $a \vdash \neg g$ and $a \not\vdash \neg g \wedge t$. Since $\neg t \vdash x$, then $a \wedge x \vdash \neg g$ and $a \wedge x \not\vdash \neg g \wedge t$. Therefore, $out_3^{wr}(N, a \wedge x) = Cn(\neg t) = out_3^{wr}(N, a)$.

6 Comparison with Constrained I/O Logic and Dyadic Deontic Logic

In this section, we compare our approach with two formalisms for CTD reasoning: constrained I/O logic [11] and dyadic deontic logic [17].

6.1 Constrained I/O Logic

Constrained I/O logic [11] is intended to deal with CTD paradoxes, which cannot be handled in unconstrained I/O logic (i.e., the output operations $out_1 - out_4$ in Section 3). In constrained I/O logic, the output operations are equipped with an additional parameter $C \subseteq \mathcal{L}$, which is to be understood as ‘‘constraints’’. Given a normative system N , an input a , and a set of constraints C , we need to define the net output $out_c(N, a, C)$. The core idea is the notion of ‘‘maxfamily’’, which is defined such that for each (unconstrained) output operation out_i (where $i \in \{1, 2, 3, 4\}$)

$maxfamily(N, a, C)$ is the set of \subseteq -maximal subsets N' of N such that
 $out_i(N', a)$ is consistent with C .

Based on the notion of ‘‘maxfamily’’, two kinds of net output operations are defined:

$$\begin{aligned} \text{Skeptical: } out_c^\cap(N, a, C) &= \bigcap_{N' \in maxfamily(N, a, C)} out_i(N', a) \\ \text{Credulous: } out_c^\cup(N, a, C) &= \bigcup_{N' \in maxfamily(N, a, C)} out_i(N', a) \end{aligned}$$

For CTD reasoning, it is assumed that $C = a$. We note that the skeptical net output operation gives the same results on Forrester’s and Chisholm’s paradoxes as our approach. That is,

Observation. Let $N = \{(\top, \neg k), (k, k \wedge g)\}$ and let out_1 be the underlying unconstrained I/O logic. Then $out_c^\cap(N, a, a) = out_1^{wr}(N, a)$ for all formulas a .

Observation. Let $N = \{(\top, g), (g, t), (\neg g, \neg t)\}$ and let out_3 be the underlying unconstrained I/O logic. Then $out_c^\cap(N, a, a) = out_3^{wr}(N, a)$ for all formulas a .

However, this correspondence does not hold in the general case, as shown in the next example. Note that this formalizes Example 2.1 in Section 2.

Example 6.1 Let $N = \{(\top, g), (\top, t), (\neg g, \neg t)\}$ and let the underlying unconstrained I/O logic be any of $out_1 - out_4$. We have:

$$\begin{aligned} out_c^\cap(N, \neg g, \neg g) &= Cn(\emptyset) \\ out_c^\cup(N, \neg g, \neg g) &= Cn(t) \cup Cn(\neg t) \end{aligned}$$

In contrast, $out_1^{wr}(N, \neg g) = Cn(\neg t)$.

To provide another example illustrating the difference between our approach and constrained I/O logic, we consider (parts of) the cottage example in Section 2:

Example 6.2 Let $N = \{(\top, \neg f), (c, f)\}$ and let out_1 be the underlying unconstrained I/O logic. We have:

$$\begin{aligned} out_c^\cap(N, c, c) &= Cn(\emptyset) \\ out_c^\cup(N, \neg g, \neg g) &= Cn(f) \cup Cn(\neg f) \\ out_1^{wr}(N, c) &= Cn(\perp) \end{aligned}$$

6.2 Dyadic Deontic Logic

In dyadic deontic logic (DDL) [17], a new dyadic operator $\bigcirc(\psi/\varphi)$ is added into the language of propositional logic, which reads as “Given φ , it is obligatory that ψ ”. Thus, $\bigcirc(\psi/\varphi)$ expresses the same concept as the pair (a, x) in I/O logic. But the meaning of the operator $\bigcirc(\psi/\varphi)$ is given by Kripke semantics. Formally, given a Kripke model $M = (W, \succeq, V)$ (in which the accessibility relation $w \succeq v$ means that “state w is at least as good as state t ”) and a state $w \in W$, then

$$M, w \models \bigcirc(\psi/\varphi) \text{ iff } best([\![\varphi]\!]) \subseteq [\![\psi]\!],$$

where $[\![\varphi]\!]$ is the truth set of φ in M and $best([\![\varphi]\!]) = \{s \in [\![\varphi]\!] \mid \forall t \in [\![\varphi]\!], s \succeq t\}$. For any set of formulas $\Gamma \cup \{\varphi\}$ in DDL, we write $\Gamma \models \varphi$ if for all models M and states w , if $M, w \models \psi$ for all $\psi \in \Gamma$, then $M, w \models \varphi$.

Dyadic deontic logic is also intended to solve CTD paradoxes like Chisholm’s. It is well known that Chisholm’s paradox can be consistently represented in DDL (see, e.g., [17]), i.e., the set of formulas $\{\bigcirc(g/\top), \bigcirc(t/g), \bigcirc(\neg t/\neg g), \neg g\}$ is satisfiable in DDL.

To compare out_1^{wr} with DDL, we first need to “translate” each property in {REF, AT, IEQ, WO, AND, wRM} into DDL. We interpret the pair (a, x) as the formula $\bigcirc(x/a)$ in DDL and for all $N \subseteq \mathcal{L} \times \mathcal{L}$, we write N^\bigcirc for the set obtained by replacing each $(a, x) \in N$ with $\bigcirc(x/a)$. We identify $(a, x) \in out(N)$ with $N^\bigcirc \models \bigcirc(x/a)$.⁴ Then we can translate, e.g., the property IEQ as follows:

$$\text{If } N^\bigcirc \models \bigcirc(x/a) \text{ and } a \dashv b, \text{ then } N^\bigcirc \models \bigcirc(x/b).$$

⁴ The translation of I/O logic into DDL can be found in [17, p. 46].

It is not hard to see that the above holds in DDL. Similarly, the translated versions of REF, AT, WO, AND also hold in DDL. However, things are different with the translated version of wRM:

If $N^\circ \not\models \bigcirc(\neg(a \wedge b)/a)$ and $N^\circ \models \bigcirc(x/a)$, then $N^\circ \models \bigcirc(x/a \wedge b)$. (*)

Observation. The above translated version of wRM does not hold in DDL.

Proof. Let $\Gamma = \{\bigcirc(x/a)\}$. We have $\Gamma \not\models \bigcirc(\neg(a \wedge b)/a)$ and $\Gamma \models \bigcirc(x/a)$. For the former, consider the following model $M = (W, \succeq, V)$:

$$W = \{w\}, \succeq = \{(w, w)\}, \text{ and } V(a) = V(b) = V(x) = \{w\}.$$

We have $M, w \models \bigcirc(x/a)$ and $M, w \not\models \bigcirc(\neg(a \wedge b)/a)$. However, $\Gamma \not\models \bigcirc(x/a \wedge b)$, because the principle SI $\bigcirc(x/a) \rightarrow \bigcirc(x/a \wedge b)$ fails in DDL. \square

Remark 6.3 The failure of (*) in DDL may seem confusing. As is known, the (RM) axiom $\neg \bigcirc(\neg b/a) \wedge \bigcirc(x/a) \rightarrow \bigcirc(x/a \wedge b)$ is valid on the class of all transitive models, and this also applies to the variation $\neg \bigcirc(\neg(a \wedge b)/a) \wedge \bigcirc(x/a) \rightarrow \bigcirc(x/a \wedge b)$. However, the latter should not be confused with (*) because (*) is a meta-property about the logical consequence relation of DDL. In particular, we interpret ‘‘negation’’ as underivability/failure in (*).

We conclude this section by noting another difference between our approach and DDL. Let N be as in Example 6.2. Then $N^\circ = \{\bigcirc(\neg f/\top), \bigcirc(f/c)\}$. In Åqvist’s G system (i.e., the consequence relation on the class of limited and transitive models, see [17]), it is the case that $\bigcirc(\neg c/\top)$ can be derived from N° . This may be considered counterintuitive. Note that in our approach, $\neg c \notin \text{out}_1^{wr}(N, \top)$.

7 Summary

There are two main approaches to deontic logic. They are either based on preference (DDL) or based on normative or rule-based systems like I/O logic. Recently, various authors have tried to connect and combine elements of both traditions. This paper fits into this trend by incorporating a key reasoning pattern, rational monotony, from DDL into I/O logic.

First, we remove the SI rule from I/O logic and present new representation results for the resulting systems, which we call $\text{out}_1^- - \text{out}_3^-$. We leave for further research the representation result for out_4^- .

Then, we investigate the extension of out_1^- with the original version of rational monotony. For this, we employ the idea of ‘‘reduction’’ from answer set programming. In the literature, there are works on embedding input/output logics into (extensions of) answer set programs, e.g., [5,8]. But our work employs answer set programming to devise new input/output logics.

The original version turns out to be unsuitable for CTD reasoning, however. Therefore, we consider a weaker form of rational monotony, i.e., the wRM property. We show that, for the extension of out_1^- with wRM, there is a stable set and it is unique. It is represented by the I/O operation out_1^{wr} . For

extensions of $out_2^- - out_4^-$ with wRM, we present sufficient conditions for the existence of stable sets.

Finally, we compare our approach to constrained I/O logic and DDL. For future research, one direction is to extend our results to other I/O logics. We may also compare them with IOL with a built-in consistency check and employ formal argumentation to use the new system for the purpose of deontic explanation.

Appendix

A Proof of Proposition 3.2

(1) Let out be an output operation such that $out(N, a) = Cn(N(Eq(a)))$. We show that $out(N)$ is the smallest set closed under {REF, AT, IEQ, WO, AND}. It is straightforward to verify that $out(N)$ is closed under the set of properties. To show that $out(N)$ is the smallest, let out' be an arbitrary output operation such that $out'(N)$ is closed under the set of properties. It suffices to show that $out(N, a) \subseteq out'(N, a)$ for all formulas a . If $x \in out(N, a) = Cn(N(Eq(a)))$, then there must be $x_1, \dots, x_n \in N(Eq(a))$ ($n \geq 0$) such that $\{x_1, \dots, x_n\} \vdash x$. To show that $x \in out'(N, a)$, it suffices to show that $x_i \in out'(N, a)$ for all i since $out'(N)$ is closed under AND and WO. For each $x_i \in N(Eq(a))$, there must be a pair $(a', x_i) \in N$ such that $a' \dashv\vdash a$. Since $out'(N)$ is closed under REF, $(a', x_i) \in out'(N)$. Since $out'(N)$ is closed under IEQ, $(a, x_i) \in out'(N)$, i.e., $x_i \in out'(N, a)$.

(3). Let out be an output operation such that $out(N, a) = \bigcup_{i \in \mathbb{N}} A_i$ where

- $A_0 = out_1^-(N, a)$;
- $A_{i+1} = Cn(A_i \cup \bigcup_{x \in A_i} out_1^-(N, a \wedge x))$.

We first show that $out(N)$ is closed under {REF, AT, IEQ, WO, AND, CT}. We show only the case for CT. Suppose $x \in out(N, a)$. We need to show that $out(N, a \wedge x) \subseteq out(N, a)$. Let $out(N, a) = \bigcup_{i \in \mathbb{N}} A_i$ and $out(N, a \wedge x) = \bigcup_{i \in \mathbb{N}} B_i$.

Then there must be an $i \in \mathbb{N}$ such that $x \in A_i$. It suffices to show that $B_k \subseteq A_{i+k+1}$ for all $k \in \mathbb{N}$. This can be done by induction on k . The base is to show that $B_0 \subseteq A_{i+1}$. Note that $B_0 = out_1^-(N, a \wedge x)$ and $A_{i+1} = Cn(A_i \cup \bigcup_{x \in A_i} out_1^-(N, a \wedge x))$. Since $x \in A_i$, it is obvious that $B_0 \subseteq A_{i+1}$. Suppose $B_k \subseteq A_{i+k+1}$, we need to show that $B_{k+1} \subseteq A_{i+k+2}$. Note that

$$\begin{aligned} B_{k+1} &= Cn(B_k \cup \bigcup_{x \in B_k} out_1^-(N, a \wedge x)); \\ A_{i+k+2} &= Cn(A_{i+k+1} \cup \bigcup_{x \in A_{i+k+1}} out_1^-(N, a \wedge x)). \end{aligned}$$

Since $B_k \subseteq A_{i+k+1}$, it is clear that $B_{k+1} \subseteq A_{i+k+2}$.

Then we have to demonstrate that $out(N)$ is the smallest. Let $out'(N)$ be any set closed under {REF, AT, IEQ, WO, AND, CT}. We show that

$\text{out}(N, a) \subseteq \text{out}'(N, a)$ for all formulas a . Since $\text{out}(N, a) = \bigcup_{i \in \mathbb{N}} A_i$, all we need to show is that $A_i \subseteq \text{out}'(N, a)$ by induction on i . The base is trivial. Suppose $A_i \subseteq \text{out}'(N, a)$. We need to show that $A_{i+1} = Cn(A_i \cup \bigcup_{x \in A_i} \text{out}_1^-(N, a \wedge x)) \subseteq \text{out}'(N, a)$. Since $\text{out}'(N)$ is closed under AND and WO, it suffices to show that for all $x \in A_i$, $\text{out}_1^-(N, a \wedge x) \subseteq \text{out}'(N, a)$. Since $x \in A_i \subseteq \text{out}'(N, a)$ and $\text{out}'(N)$ is closed under CT, $\text{out}'(N, a \wedge x) \subseteq \text{out}'(N, a)$. Note that $\text{out}_1^-(N, a \wedge x) \subseteq \text{out}'(N, a \wedge x)$. Thus, $\text{out}_1^-(N, a \wedge x) \subseteq \text{out}'(N, a)$.

B Proof in Example 5.12

- $\text{out}_3^{wr}(N, a) = Cn(g \wedge t)$ if $a \not\vdash \neg g \vee \neg t$. This can be shown by induction on the relation \prec . If $a \not\vdash \top$, then $\text{out}_3^{wr}(N, a) = \text{out}_3^-(N, \top) = Cn(g \wedge t)$. Suppose $\text{out}_3^{wr}(N, b) = Cn(g \wedge t)$ for all $b \prec a$ with $b \not\vdash \neg g \vee \neg t$. Note that

$$\text{out}_3^{wr}(N, a) = Cn \left(\text{out}_3^-(N, a) \cup \bigcup_{\{b | b \prec a \wedge \neg a \notin \text{out}_3^{wr}(N, b)\}} \text{out}_3^{wr}(N, b) \right).$$

Since $a \not\vdash \neg g$, then $\text{out}_3^-(N, a) \subseteq Cn(g \wedge t)$. Since $\neg a \notin \text{out}_3^{wr}(N, \top) = Cn(g \wedge t)$, $\text{out}_3^{wr}(N, \top) \subseteq \text{out}_3^{wr}(N, a)$. Therefore, $\text{out}_3^{wr}(N, a) = Cn(g \wedge t)$.

- $\text{out}_3^{wr}(N, a) = Cn(\emptyset)$ if $a \vdash \neg g \vee \neg t$ and $a \not\vdash \neg g$. We show this by induction on the relation \prec . Note that $\text{out}_3^{wr}(N, \neg g \vee \neg t) = \text{out}_3^-(N, \neg g \vee \neg t) = Cn(\emptyset)$. For the inductive step, note that

$$\text{out}_3^{wr}(N, a) = Cn \left(\text{out}_3^-(N, a) \cup \bigcup_{\{b | b \prec a \wedge \neg a \notin \text{out}_3^{wr}(N, b)\}} \text{out}_3^{wr}(N, b) \right).$$

Since $a \not\vdash \top$, $a \not\vdash g$, and $a \not\vdash \neg g$, then $\text{out}_3^-(N, a) = Cn(\emptyset)$. For each $b \prec a$ with $\neg a \notin \text{out}_3^{wr}(N, b)$, we have $b \vdash \neg g \vee \neg t$ and $b \not\vdash \neg g$. Therefore, by IH, $\text{out}_3^{wr}(N, b) = Cn(\emptyset)$. Thus, $\text{out}_3^{wr}(N, a) = Cn(\emptyset)$.

- $\text{out}_3^{wr}(N, a) = Cn(\neg t)$ if $a \vdash \neg g$ and $a \not\vdash \neg g \wedge t$. We show this by induction on the relation \prec . Note that $\text{out}_3^{wr}(N, \neg g) = \text{out}_3^-(N, \neg g) = Cn(\neg t)$. For the inductive step, note that

$$\text{out}_3^{wr}(N, a) = Cn \left(\text{out}_3^-(N, a) \cup \bigcup_{\{b | b \prec a \wedge \neg a \notin \text{out}_3^{wr}(N, b)\}} \text{out}_3^{wr}(N, b) \right).$$

Since $a \not\vdash \top$ and $a \not\vdash g$, then $\text{out}_3^-(N, a) \subseteq Cn(\neg t)$. For each $b \prec a$ with $\neg a \notin \text{out}_3^{wr}(N, b)$, we have $b \vdash \neg g \vee \neg t$ and $b \not\vdash \neg g \wedge t$. Therefore, $\text{out}_3^{wr}(N, b) \subseteq Cn(\neg t)$. Thus, $\text{out}_3^{wr}(N, a) \subseteq Cn(\neg t)$. Since $\neg a \notin \text{out}_3^{wr}(N, \neg g) = Cn(\neg t)$, then $Cn(\neg t) \subseteq \text{out}_3^{wr}(N, a)$. Therefore, $\text{out}_3^{wr}(N, a) = Cn(\neg t)$.

- $\text{out}_3^{wr}(N, a) = Cn(\emptyset)$ if $a \vdash \neg g \wedge t$. It is easy to see that $\text{out}_3^-(N, a) = Cn(\emptyset)$. Besides, for each $b \prec a$ with $\neg a \notin \text{out}_3^{wr}(N, b)$, we have $\text{out}_3^{wr}(N, b) = Cn(\emptyset)$.

References

- [1] Chisholm, R. M., *Contrary-to-duty imperatives and deontic logic*, Analysis **24** (1963), pp. 33–36.
URL <http://www.jstor.org/stable/3327064>
- [2] Forrester, J. W., *Gentle murder, or the adverbial samaritan*, The Journal of Philosophy **81** (1984), pp. 193–197.
URL <http://www.jstor.org/stable/2026120>
- [3] Gelfond, M. and V. Lifschitz, *The stable model semantics for logic programming*, in: R. Kowalski, Bowen and Kenneth, editors, *Proceedings of International Logic Programming Conference and Symposium* (1988), pp. 1070–1080.
URL <http://www.cs.utexas.edu/users/ai-lab?gel188>
- [4] Gelfond, M. and V. Lifschitz, *Classical negation in logic programs and disjunctive databases*, New generation computing **9** (1991), pp. 365–385.
- [5] Gonçalves, R. and J. J. Alferes, *An embedding of input-output logic in deontic logic programs*, in: T. Ågotnes, J. Broersen and D. Elgesem, editors, *Deontic Logic in Computer Science* (2012), pp. 61–75.
- [6] Hansson, B., “An Analysis of Some Deontic Logics,” Springer Netherlands, Dordrecht, 1971 pp. 121–147.
- [7] Horty, J. F., *Deontic logic as founded on nonmonotonic logic*, Annals of Mathematics and Artificial Intelligence **9** (1993), pp. 69–91.
- [8] José Júlio Alferes, R. G. and J. Leite, *Equivalence of defeasible normative systems*, Journal of Applied Non-Classical Logics **23** (2013), pp. 25–48.
URL <https://doi.org/10.1080/11663081.2013.798996>
- [9] Lehmann, D. and M. Magidor, *What does a conditional knowledge base entail?*, Artificial Intelligence **55** (1992), pp. 1–60.
URL <https://www.sciencedirect.com/science/article/pii/000437029290041U>
- [10] Makinson, D. and L. Van Der Torre, *Input/output logics*, Journal of philosophical logic **29** (2000), pp. 383–408.
- [11] Makinson, D. and L. van der Torre, *Constraints for input/output logics*, Journal of Philosophical Logic **30** (2001), pp. 155–185.
URL <https://doi.org/10.1023/A:1017599526096>
- [12] Makinson, D. and L. van der Torre, *Permission from an input/output perspective*, Journal of Philosophical Logic **32** (2003), pp. 391–416.
URL <https://doi.org/10.1023/A:1024806529939>
- [13] Makinson, D. and L. van der Torre, “What is Input/Output Logic?” Springer Netherlands, Dordrecht, 2003 pp. 163–174.
URL https://doi.org/10.1007/978-94-017-0395-6_12
- [14] Prakken, H. and M. Sergot, *Contrary-to-duty obligations*, Studia Logica **57** (1996), pp. 91–115.
- [15] Aqvist, L., “An introduction to deontic logic and the theory of normative systems,” Humanities Press, 1988.
- [16] van der Torre, L., “Reasoning about Obligations: Defeasibility in Preference-based Deontic Logic,” Ph.D. thesis, Erasmus University Rotterdam (1997).
- [17] van der Torre, L. and X. Parent, “Introduction to Deontic Logic and Normative Systems,” College Publications, 2018.
- [18] van der Torre, L. and X. Parent, *Detachment in normative systems: Examples, inference patterns, properties*, Journal of Applied Logics **9** (2022), pp. 1087–1130.
URL <https://collegepublications.co.uk/ifcolog/?00056>
- [19] Van Der Torre, L. W. N. 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 - Volume 2*, IJCAI'95 (1995), p. 1525–1532.
- [20] Van Fraassen, B. C., *Values and the heart's command*, The Journal of Philosophy **70** (1973), pp. 5–19.