From Input/Output Logics to Conditional Logics via Sequents – With Provers (extended version)[⋆]

Björn Lellmann[0000−0002−5335−1838]

TU Wien, Vienna, Austria
lellmann@logic.at

Abstract. We consider cut-free sequent calculi for a number of deontic logics from the family of Input/Output logics. These sequent calculi provide a correspondence to the flat fragment of certain conditional logics. Two of the introduced calculi are non-standard in that they include non-derivability statements, and hence are interesting also from a purely technical perspective. We further modularise the calculi in an extended sequent framework. Proof search in the extended calculi is implemented in Prolog, providing seemingly the first automated reasoning systems for some of the considered logics.

Keywords: I/O Logic · Conditional Logic · Deontic Logic · Sequent systems

1 Introduction

A formalism which has recently gained interest in the field of deontic logic is that of *Input/Output logics* [16,23]. Here, conditional obligations such as "If there is a dog, then there must be a fence" are treated as *Input-Output* pairs, intuitively converting their input (the conditions under which the conditional obligation holds, e.g., "there is a dog") into their output (what is obligatory under these conditions, e.g., "there is a fence"). In the Input/Output approach this conversion mechanism, called *detachment*, is taken as the core mechanism of deontic logics, and is used to analyse phenomena and problems of deontic logic including, e.g., Contrary-to-Duty reasoning (reasoning with and about violated norms) or deontic paradoxa and dilemmas. In this framework, an Input/Output logic is viewed as a "transformation engine", which converts an input, i.e., a state description, into an output, i.e., what should be the case, using a set of conditional obligations in the form of Input/Output pairs. As a main aspect of the Input/Output framework, the Input/Output pairs are given by a *meta-level* connective instead of an *object-level* connective as in, e.g., dyadic deontic logic or conditional logic. Different Input/Output logics are then obtained (on the syntactical side) by varying the mechanisms of obtaining new input-output pairs from a given set of these pairs.

[⋆] This work has been supported by BRISE-Vienna (UIA04-081), a European Union Urban Innovative Actions project

While the more theoretical side of the basic Input/Output logics by now is rather well understood, their automated reasoning side has not yet been fully explored: The only more practical approaches in this direction so far seem to be the semantical embedding of some systems of Input/Output logic into Higher-Order logic enabling automation for these systems in [3] and a goal-directed method for deciding certain Input/Output logics introduced in [29]. However, the embedding into Higher-Order Logic makes use of an embedding of Input/Output logics into certain modal logics, and the goal-directed decision procedures are based heavily on the semantic characterisation of the logics.

Here we take a more proof-theoretic approach and exploit a strong similarity to the KLM framework for nonmonotonic reasoning [10]. Analogously to the notion of an Input/Output pair the KLM framework is based on a meta-level connective for *nonmonotonic inference*, written $\Gamma \mid\!\sim A$ and interpreted as "Γ nonmonotonically entails A". Different systems then are given by different rules for this connective. However, it has been observed already in *op. cit.*, that this meta-level connective corresponds to a dyadic object-level connective, specifically that of *conditional logics*, and that different systems in the KLM framework therefore correspond to the flat (i.e., unnested) fragment of various conditional logics. This of course opens up the possibility of transferring certain results from one framework to the other. In addition, the formulation in terms of an object-level connective facilitates the application of syntactic methods, in particular the construction and use of sequent systems.

In the present paper we will use the same idea to obtain axiomatisations for the logical connective corresponding to input/output pairs in certain Input/Output logics. With the aim of obtaining automated reasoning procedures we consider corresponding sequent systems for these logics. These correspondences are also interesting in their own right, because they yield a representation of certain Input/Output logics in conditional logics, resulting in an alternative semantics. Two of the sequent systems are in addition non-standard in that they mention underivability in the premisses, stemming from the fact that the corresponding Input/Output logics contain *consistency constraints* in the formulation of the rules. With respect to automated reasoning, we then consider a modification of the sequent systems which facilitates a prototype implementation.

Of course the idea of turning Input/Output pairs into logical connectives goes against the original idea of treating conditional obligations expressly at the meta-level [16]. While there are certainly good philosophical arguments to do so, here we do not take a stance on this matter and treat the connectives from a purely syntactical point of view.

2 Input/Output Logics And Their Sequent Calculi

We briefly recall the relevant Input/Output logics, henceforth also simply *I/O logics*, and then consider their sequent calculi. The reader is referred to [16,23] for more details on Input/Output logics including the semantics, motivation and philosophical discussion. We defer the technical results about the calculi to Sec. 3.

$$\frac{(A,X) \quad B \vdash A}{(B,X)}\ \mathsf{SI} \qquad \frac{(A,X) \quad X \vdash Y \quad Y \vdash X}{(A,Y)}\ \mathsf{OEQ} \qquad \frac{(A,X) \quad X \vdash Y}{(A,Y)}\ \mathsf{WO}$$

$$\frac{}{(\top,\top)}\ \top \qquad \frac{}{(A,A)}\ \mathsf{ID} \qquad \frac{(A,X) \quad (A \wedge X, Y)}{(A,Y)}\ \mathsf{CT}$$

$$\frac{(A,X) \quad (A,Y)}{(A, X \wedge Y)}\ \mathsf{AND} \qquad \frac{(A,X) \quad (A,Y) \quad A \wedge X \wedge Y \text{ consistent}}{(A, X \wedge Y)}\ \mathsf{RAND}$$

$$\frac{(A,X) \quad (A \wedge X, Y)}{(A, X \wedge Y)}\ \mathsf{ACT} \qquad \frac{(A,X) \quad (A \wedge X, Y) \quad A \wedge X \wedge Y \text{ consistent}}{(A, X \wedge Y)}\ \mathsf{RACT}$$

Fig. 1. I/O logic rules

The set Prop of *propositional formulae* is given as usual by the grammar $\mathsf{Prop} ::= \mathcal{V} \mid \bot \mid \top \mid \neg\mathsf{Prop} \mid \mathsf{Prop} \wedge \mathsf{Prop} \mid \mathsf{Prop} \vee \mathsf{Prop} \mid \mathsf{Prop} \to \mathsf{Prop}$ where $\mathcal{V}$ is a countable infinite set of *propositional variables*. We assume the usual conventions about binding strength of the operators, i.e., $\neg$ binds stronger than $\wedge$ binds stronger than $\vee$ binds stronger than $\to$.

Definition 1. *An* Input/Output pair*, short* I/O pair *is a tuple* (A, X)*, where* $A, X \in \mathsf{Prop}$ *are propositional formulae.*

In the following we only consider mainly *unconstrained* I/O logics (see, e.g., [23] for the details). From the purely syntactic point of view, these logics then are given by different rules for obtaining new I/O pairs from a given set of I/O pairs, captured in the form of different derivability relations. Here we consider the following I/O logics: *simple-minded output* deriv_1, *simple-minded throughput* deriv_1^+, *reusable output* deriv_3, and *reusable throughput* deriv_3^+ from [16], *simple-minded output without weakening* (or aggregative simple-minded output) $\mathsf{ag\_der}_1$ and *reusable output without weakening* (or aggregative reusable output) $\mathsf{ag\_der}_3$ from [30,26], and *simple minded output with consistency check* $\mathsf{c\_ag\_der}_1$ as well as *basic output with consistency check* $\mathsf{c\_ag\_der}_3$ from [24,25].

The rules are given in Fig. 1. Rule SI (*Strengthening of the Input*) corresponds to downwards monotonicity in the first argument of the pair operator, while WO (*Weakening of the Output*) corresponds to upwards monotonicity in the second argument. Rule OEQ (*Output Equivalence*) is the weaker version of the latter stating congruence in the second argument. Rules $\top$ (*Tautology*) and ID (*Identity*) are self-explanatory. Rule AND and its weaker version RAND (*Restricted AND*) state that conjunction distributes over the second argument, while the rules CT (*Cumulative Transitivity*), ACT (*Aggregative Cumulative Transitivity*) and RACT (*Restricted Aggregative Cumulative Transitivity*) state various weaker versions of transitivity. Note that since there is no nesting of the I/O pair operator, the entailment relation in the rules $\mathsf{SI}, \mathsf{OEQ}$ and WO as well as the consistency requirement in the rules RAND and RACT range over classical propositional logic.

Logic	SI	OEQ	WO	$\top$	ID	RAND	AND	RACT	ACT	CT	Reference
deriv_1	✓	(✓)	✓	✓		(✓)	✓				[16]
deriv_1^+	✓	(✓)	✓	(✓)	✓	(✓)	✓				[16]
deriv_3	✓	(✓)	✓	✓		(✓)	✓	(✓)	(✓)	✓	[16]
deriv_3^+	✓	(✓)	✓	(✓)	✓	(✓)	✓	(✓)	(✓)	✓	[16]
$\mathsf{ag\_der}_1$	✓	✓				(✓)	✓				$\mathcal{D}_1$ in [26]
$\mathsf{ag\_der}_3$	✓	✓				(✓)	✓	(✓)	✓		$\mathcal{D}_3$ in [26]
$\mathsf{c\_ag\_der}_1$	✓	✓				✓					D_1 in [25]
$\mathsf{c\_ag\_der}_3$	✓	✓				(✓)		✓			D_3 in [25]

Fig. 2. The different I/O logics. Checkmarks in parentheses are implied.

$$\frac{}{\Gamma, \bot \Rightarrow \Delta}\ \bot_L \qquad \frac{}{\Gamma \Rightarrow \top, \Delta}\ \top_R \qquad \frac{}{\Gamma, p \Rightarrow p, \Delta}\ \mathsf{init} \qquad \frac{\Gamma \Rightarrow A, \Delta}{\Gamma, \neg A \Rightarrow \Delta}\ \neg_L \qquad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \neg A, \Delta}\ \neg_R$$

$$\frac{\Gamma, B \Rightarrow \Delta \quad \Gamma \Rightarrow A, \Delta}{\Gamma, A \to B \Rightarrow \Delta}\ \to_L \qquad \frac{\Gamma, A \Rightarrow \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \vee B \Rightarrow \Delta}\ \vee_L \qquad \frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta}\ \wedge_L$$

$$\frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \to B, \Delta}\ \to_R \qquad \frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \vee B, \Delta}\ \vee_R \qquad \frac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta}\ \wedge_R$$

Fig. 3. The classical propositional sequent rules

Definition 2. *Let G be a set of I/O pairs, and let $\mathcal{L}$ be one of* deriv_1*,* deriv_1^+*,* deriv_3*,* deriv_3^+*,* $\mathsf{ag\_der}_1$*,* $\mathsf{ag\_der}_3$*. An I/O pair (A, X) is* derivable from *G in $\mathcal{L}$, written $G \vdash_{\mathcal{L}} (A, X)$, iff there is a derivation in $\mathcal{L}$ with conclusion (A, X) whose leaves are I/O pairs from G or entailment or consistency statements true in classical propositional logic. Here as usual a* derivation in *$\mathcal{L}$ is a finite labelled directed tree, whose nodes are labelled with I/O pairs or statements about propositional logic such that the label of each node follows from the labels of its children using the rules of $\mathcal{L}$ as given in Fig. 2. For $\mathcal{L}$ one of* $\mathsf{c\_ag\_der}_1$ *or* $\mathsf{c\_ag\_der}_3$ *the definition is the same with the additional requirement that for all leaves (B, Y) of the derivation we have that $B \wedge Y$ is consistent.*

In order to formulate the sequent systems corresponding to such logics we internalise the I/O pairs using a corresponding logical connective $>$ as follows.

Definition 3. *The set $\mathcal{F}$ of formulae in the* internalised I/O language *is given by $\mathcal{F} ::= \mathcal{V} \mid \bot \mid \top \mid \mathcal{F} \wedge \mathcal{F} \mid \mathcal{F} \vee \mathcal{F} \mid \mathcal{F} \to \mathcal{F} \mid \mathcal{F} > \mathcal{F}$. A* sequent *in the internalised I/O language is a tuple of multisets of formulae from $\mathcal{F}$, written $\Gamma \Rightarrow \Delta$.*

We assume that all propositional connectives bind stronger than $>$. The sequent systems considered below all contain the standard $\mathsf{G3p}$ rules for classical propositional logic given in Fig. 3 (see also [32]).

Converting the rules for the I/O pairs into sequent rules by simply moving the I/O pairs from the premisses to the antecedent of the conclusion yields the

$$\frac{B \Rightarrow A}{(A > X) \Rightarrow (B > X)}\ (\mathsf{SI}) \quad \frac{X \Rightarrow Y \quad Y \Rightarrow X}{(A > X) \Rightarrow (A > Y)}\ (\mathsf{OEQ}) \quad \frac{X \Rightarrow Y}{(A > X) \Rightarrow (A > Y)}\ (\mathsf{WO})$$

$$(\top) : \Rightarrow \top > \top \qquad (\mathsf{ID}) : \Rightarrow (A > A) \qquad (\mathsf{CT}) : \ (A > X) \wedge (A \wedge X > Y) \Rightarrow (A > Y)$$

$$(\mathsf{AND}) : \ (A > X) \wedge (A > Y) \Rightarrow (A > X \wedge Y)$$

$$(\mathsf{ACT}) : (A > X) \wedge (A \wedge X > Y) \Rightarrow (A > X \wedge Y)$$

$$\frac{A \wedge X \wedge Y \text{ consistent}}{(A > X) \wedge (A > Y) \Rightarrow (A > X \wedge Y)}\ (\mathsf{RAND})$$

$$\frac{A \wedge X \wedge Y \text{ consistent}}{(A > X) \wedge (A \wedge X > Y) \Rightarrow (A > X \wedge Y)}\ (\mathsf{RACT})$$

Fig. 4. Axioms and rules corresponding to the I/O rules

rules and axiomatic sequents in Fig. 4. Of course, replacing the sequent arrow in these with an implication yields Hilbert-style axiomatisations. Using the methods of [27,12,11] to absorb cuts between the conclusions of these rules into the rule set yields the rules in Fig. 5. To save space in the presentation of the rules we abuse notation and use set notation in the premisses. E.g., instead of writing

$$\frac{C \Rightarrow A_1 \quad C, B_1 \Rightarrow A_2 \quad C, B_1, B_2 \Rightarrow A_3 \quad B_1, B_2, B_3 \Rightarrow D}{(A_1 > B_1), (A_2 > B_2), (A_3 > B_3) \Rightarrow (C > D)}\ \mathsf{R}_3$$

we write

$$\frac{\{C, B_1, \ldots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq 3\} \quad B_1, B_2, B_3 \Rightarrow D}{(A_1 > B_1), (A_2 > B_2), (A_3 > B_3) \Rightarrow (C > D)}\ \mathsf{R}_3 \ .$$

Note also that the systems include the special case of the rules for $n = 0$, i.e., the rules

$$\frac{\Rightarrow D}{\Gamma \Rightarrow (C > D), \Delta}\ \mathsf{CC}_0 \qquad \frac{C \Rightarrow D}{\Gamma \Rightarrow (C > D), \Delta}\ \mathsf{CCI}_0$$

which are the same as R_0 and RI_0, respectively. Most significantly, the rules $\mathsf{c\_ag\_CC}_n$ and $\mathsf{c\_ag\_R}_n$ include *underivability statements* of the form $\not\vdash \Gamma \Rightarrow \Delta$ in the premisses. In order to capture this we use the following notion of derivability:

Definition 4. *A* proto-derivation for a sequent $\Gamma \Rightarrow \Delta$ in $\mathsf{G}_\mathcal{L}$ *is a finite directed labelled tree, where the root is labelled with $\Gamma \Rightarrow \Delta$ and:*

- *each internal node is labelled with a sequent, each leaf is labelled with a sequent or an* underivability statement *of the form $\not\vdash \Sigma \Rightarrow \Pi$*
- *whenever a node has a standard sequent as its label, then that sequent follows from the labels of the node's children using the rules of $\mathsf{G}_\mathcal{L}$*

The depth *of a proto-derivation is the depth of the underlying tree. A proto-derivation in $\mathsf{G}_\mathcal{L}$ is* valid *if for every underivability statement $\not\vdash \Sigma \Rightarrow \Pi$ there is no valid proto-derivation of $\Sigma \Rightarrow \Pi$ in $\mathsf{G}_\mathcal{L}$. A sequent is* derivable *in $\mathsf{G}_\mathcal{L}$, written $\vdash_{\mathsf{G}_\mathcal{L}} \Gamma \Rightarrow \Delta$ if there is a valid proto-derivation for it in $\mathsf{G}_\mathcal{L}$.*

$$\frac{\{C \Rightarrow A_i : 1 \leq i \leq n\} \quad B_1, \dots, B_n \Rightarrow D}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta}\ \mathsf{CC}_n$$

$$\frac{\{C \Rightarrow A_i : 1 \leq i \leq n\} \quad B_1, \dots, B_n, C \Rightarrow D}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta}\ \mathsf{CCI}_n$$

$$\frac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq n\} \quad B_1, \dots, B_n \Rightarrow D}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta}\ \mathsf{R}_n$$

$$\frac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq n\} \quad B_1, \dots, B_n, C \Rightarrow D}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta}\ \mathsf{RI}_n$$

$$\frac{\{C \Rightarrow A_i : 1 \leq i \leq n\} \quad \{D \Rightarrow B_i : 1 \leq i \leq n\} \quad B_1, \dots, B_n \Rightarrow D}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta}\ \mathsf{ag\_CC}_n$$

$$\frac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq n\} \quad \{D \Rightarrow B_i : 1 \leq i \leq n\} \quad B_1, \dots, B_n \Rightarrow D}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta}\ \mathsf{ag\_R}_n$$

$$\frac{\begin{array}{c}\{C \Rightarrow A_i : 1 \leq i \leq n\} \qquad \{D \Rightarrow B_i : 1 \leq i \leq n\} \\ B_1, \dots, B_n \Rightarrow D \qquad \nvdash C, D \Rightarrow\end{array}}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta}\ \mathsf{c\_ag\_CC}_n$$

$$\frac{\begin{array}{c}\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq n\} \qquad \{D \Rightarrow B_i : 1 \leq i \leq n\} \\ B_1, \dots, B_n \Rightarrow D \qquad \nvdash C, D \Rightarrow\end{array}}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta}\ \mathsf{c\_ag\_R}_n$$

$\mathsf{G}_{\mathsf{deriv}_1}$	: $\{\mathsf{CC}_n : n \geq 0\}$	$\mathsf{G}_{\mathsf{deriv}_3}$	: $\{\mathsf{R}_n : n \geq 0\}$
$\mathsf{G}_{\mathsf{deriv}_1^+}$	: $\{\mathsf{CCI}_n : n \geq 0\}$	$\mathsf{G}_{\mathsf{deriv}_3^+}$	: $\{\mathsf{RI}_n : n \geq 0\}$
$\mathsf{G}_{\mathsf{ag\_der}_1}$	: $\{\mathsf{ag\_CC}_n : n \geq 1\}$	$\mathsf{G}_{\mathsf{ag\_der}_3}$	: $\{\mathsf{ag\_R}_n : n \geq 1\}$
$\mathsf{G}_{\mathsf{c\_ag\_der}_1}$	: $\{\mathsf{c\_ag\_CC}_n : n \geq 1\}$	$\mathsf{G}_{\mathsf{c\_ag\_der}_3}$	: $\{\mathsf{c\_ag\_R}_n : n \geq 1\}$

Fig. 5. Sequent rules for I/O logics

Of course, since the definition of a valid proto-derivation makes use of the very same notion, we need to show that the concept is well defined.

Lemma 5. *Every proto-derivation in $\mathsf{G}_{\mathcal{L}}$ is valid or not valid, but not both.*

Proof. By induction on the *modal rank* of its conclusion $\Gamma \Rightarrow \Delta$, i.e., the maximal nesting depth of the modal operator $>$ in a formula in the sequent. All rules of $\mathsf{G}_{\mathcal{L}}$ have the *subformula property*, i.e., every formula occurring in its premisses is a subformula of a formula occurring in its conclusion. Hence, if the modal rank of the conclusion of a proto-derivation is 0, then only propositional rules occur in it, and hence no underivability statements. Thus it is automatically valid.

Suppose that the modal rank of the conclusion of a proto-derivation $\mathcal{D}$ is $n+1$. Then again by the subformula property and the fact that the underivability statements in the rules have strictly lower modal rank than their conclusions we obtain that all the underivability statements in $\mathcal{D}$ have modal rank at most n. By induction hypothesis every proto-derivation for the sequent in such an underivability statement is either valid or not valid, but not both. Thus for every sequent occurring in such an underivability statement either there is a valid proto-derivation or there is not, but not both. Hence the proto-derivation $\mathcal{D}$ is either valid or not, but not both. $\square$

Since this definition of derivability is rather non-standard, some remarks are in order. First for the calculi without underivability statements the definition collapses to the standard notion of derivability in sequent systems. Hence the reader not interested in the calculi for $\mathsf{c\_ag\_der}_1$ and $\mathsf{c\_ag\_der}_3$ may mentally substitute the standard definition for the rest of the paper.

We could avoid underivability statements by considering an *antisequent calculus* for underivability along the lines of [4]. While for propositional rules this works well due to invertibility, for the modal rules we would need rules stating that for every modal rule which could have been used to derive a sequent at least one of its premisses is underivable. Since every ordered subset of modal formulae in a sequent corresponds to such a possible rule application, the number of premisses for the modal antisequent rules would become rather large. For the sake of a more compact presentation we therefore keep to the current formulation.

3 Technical Results And Correspondence

We now consider the properties of our calculi, starting with a number of standard results leading up to cut elimination and the correspondence to the I/O logics.

Lemma 6 (Generalised Initial Sequents). *Let $\mathcal{L}$ be one of the logics without consistency check. Then $\vdash_{\mathsf{G}_{\mathcal{L}}} \Gamma, A \Rightarrow A, \Delta$. If $\mathcal{L}$ is one of* $\mathsf{c\_ag\_der}_1$ *and* $\mathsf{c\_ag\_der}_3$ *then $\vdash_{\mathsf{G}_{\mathcal{L}}} \Gamma, A \Rightarrow A, \Delta$ for purely propositional A.*

Proof. By induction on the complexity of the formula A. In case A is of the form $C > D$ we use the modal rule with exactly one principal formula on the left hand side, e.g., in the calculus $\mathsf{G}_{\mathsf{deriv}_3^+}$ we have an application of the rule RI_1 with conclusion $(C > D) \Rightarrow (C > D)$ and premisses $C \Rightarrow C$ and $D, C \Rightarrow D$. The premisses are derivable by induction hypothesis. $\square$

Note that derivability of the generalised initial sequents does not hold unrestrictedly for the logics $\mathsf{c\_ag\_der}_1$ and $\mathsf{c\_ag\_der}_3$, because we cannot derive sequents $(A > B) \Rightarrow (A > B)$ where A and B are inconsistent. This includes examples such as $(\bot > \bot) \Rightarrow (\bot > \bot)$ or $(A > \neg A) \Rightarrow (A > \neg A)$. For the purpose of showing equivalence to I/O logics the form restricted to propositional formulae is enough, though, since I/O logics do not contain nested I/O pairs.

Lemma 7 (Invertibility of the propositional rules). *Let $\mathcal{L}$ be one of the logics considered. Then the propositional rules are* depth-preserving invertible*, i.e., whenever their conclusion is derivable with a proto-derivation of depth n, then so are their premisses.*

Proof. By induction on the depth of the proto-derivation, using the fact that the formulae with a propositional connective at the top level occur in the modal rules only as context formulae. $\square$

Lemma 8 (Admissibility of the structural rules). *Let $\mathcal{L}$ be one of the logics considered. Then the structural rules of weakening, left contraction and right contraction below are* depth-preserving admissible*, i.e, whenever their premiss is derivable in depth n, then so is their conclusion.*

$$\frac{\Gamma \Rightarrow \Delta}{\Sigma, \Gamma \Rightarrow \Delta, \Pi}\ \mathsf{W} \qquad \frac{\Gamma, A, A \Rightarrow \Delta}{\Gamma, A \Rightarrow \Delta}\ \mathsf{ICL} \qquad \frac{\Gamma \Rightarrow A, A, \Delta}{\Gamma \Rightarrow A, \Delta}\ \mathsf{ICR}$$

Proof. By induction on the depth of the proto-derivation. In case the last applied rule is a propositional rule with the contracted formula principal, as usual we use depth-preserving invertibility of the propositional rules (Lem. 7). In case the last applied rule is a modal rule with both instances of the contracted formula principal, we apply contraction to the premisses followed by the version of the same modal rule with one principal formula less. Note that the modal rules only have one principal formula on the right hand side, hence we do not need to consider contractions between principal formulae on the right and avoid having to deal with contractions in the underivability statements. □

Theorem 9 (Cut Admissibility). *The cut rule is admissible in $\mathsf{G}_{\mathcal{L}}$, i.e.,*

$$\textit{if } \vdash_{\mathsf{G}_{\mathcal{L}}} \Gamma \Rightarrow \Delta, A \textit{ and } \vdash_{\mathsf{G}_{\mathcal{L}}} A, \Sigma \Rightarrow \Pi \textit{ then } \vdash_{\mathsf{G}_{\mathcal{L}}} \Gamma, \Sigma \Rightarrow \Delta, \Pi\ .$$

Proof. As usual by double induction on the complexity of the cut formula A and the sum of the depths of the valid proto-derivations $\mathcal{D}_1$ of $\Gamma \Rightarrow \Delta, A$ and $\mathcal{D}_2$ of $A, \Sigma \Rightarrow \Pi$, see, e.g., [32]. The case where the complexity of A is 1, i.e., A is a propositional variable is standard, permuting the cut over the last applied rule in $\mathcal{D}_1$ using the induction hypothesis until that rule is init, then absorbing it into the last applied rule in $\mathcal{D}_2$. In case the complexity of A is $n+1$ we distinguish cases according to the topmost connective of A. In the propositional case we follow the standard approach and use invertibility of the propositional rules (Lem. 7) to reduce the cut to cuts on formulae of smaller complexity, potentially followed by admissibility of Contraction (Lem. 8) to eliminate duplicate formulae.

The interesting case is where A is of the form $(C > D)$. Again we permute the cut over the last applied rule in $\mathcal{D}_1$ until the cut formula is principal there using the inner induction hypothesis, then do the same with $\mathcal{D}_2$. What is left to check is that cuts between the principal formulae of two modal rules can be reduced to cuts of smaller complexity. We only consider a complicated case here, the remaining cases are similar but simpler. The full list can be found in Appendix A.1.

Suppose that $\mathcal{L}$ is $\mathsf{c\_ag\_der}_3$, and the last applied rules were $\mathsf{c\_ag\_R}_n$:

$$\frac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq n\} \quad \{D \Rightarrow B_i : i \leq n\} \quad B_1, \dots, B_n \Rightarrow D \quad \nvdash C, D \Rightarrow}{(A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D)}$$

and $\mathsf{c\_ag\_R}_m$:

$$\frac{\begin{array}{l}\{G, F_1, \ldots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad G, F_1, \ldots, F_{k-1} \Rightarrow C \\ \quad\quad \{G, F_1, \ldots, F_{k-1}, D, F_{k+1}, \ldots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \quad\quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad\quad H \Rightarrow D \\ \quad\quad\quad\quad\quad F_1, \ldots, F_{k-1}, D, F_{k+1}, \ldots, F_m \Rightarrow H \\ \quad\quad\quad\quad\quad\quad \not\vdash G, H \Rightarrow\end{array}}{(E_1 > F_1), \ldots, (E_{k-1} > F_{k-1}), (C > D), (E_{k+1} > F_{k+1}), \ldots, (E_m > F_m) \Rightarrow (G > H)}$$

Applying the induction hypothesis to cut on the formulae C and D we obtain:

$$\begin{array}{l}\{G, F_1, \ldots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad \{G, F_1, \ldots, F_{k-1}, B_1, \ldots, B_{i-1} \Rightarrow A_i : i \leq n\} \\ \quad\quad \{G, F_1, \ldots, F_{k-1}, B_1, \ldots, B_n, F_{k+1}, \ldots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \quad\quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad\quad \{H \Rightarrow B_i : i \leq n\} \\ \quad\quad\quad\quad\quad F_1, \ldots, F_{k-1}, B_1, \ldots, B_n, F_{k+1}, \ldots, F_m \Rightarrow H \\ \quad\quad\quad\quad\quad\quad \not\vdash G, H \Rightarrow\end{array}$$

and applying $\mathsf{c\_ag\_R}_{n+m-1}$ yields the desired $(E_1 > F_1), \ldots, (E_{k-1} > F_{k-1})$, $(A_1 > B_1), \ldots, (A_n > B_n), (E_{k+1} > F_{k+1}), \ldots (E_m > F_m) \Rightarrow (G > H)$. □

As usual, one of the main consequences of cut admissibilty is consistency:

Corollary 10 (Consistency). *The calculi are consistent, i.e.,* $\not\vdash_{\mathsf{G}_\mathcal{L}} \Rightarrow \bot$.

Proof. All rules have the subformula property, and no rule introduces $\bot$. □

Lemma 11 (Derivability of the axioms). *Let $\mathcal{L}$ be one of the considered logics. Then the axioms and rules of Fig. 4 for the corresponding I/O rules are derivable in $\mathsf{G}_\mathcal{L}$, restricted to non-nested formulae for $\mathsf{c\_ag\_der}_1$ and $\mathsf{c\_ag\_der}_3$.*

Proof. The rules (SI) and (WO) are special cases of the rules CC_1, CCI_1, R_1 and RI_1 respectively. The rule (OEQ) is a special case of $\mathsf{ag\_CC}_1$ and $\mathsf{ag\_R}_1$. For (RACT) we have for purely propositional formulae A, B, C:

$$\dfrac{\dfrac{A \Rightarrow A \quad A, B \Rightarrow A \wedge B \quad B \wedge C \Rightarrow B \quad B \wedge C \Rightarrow C \quad B, C \Rightarrow B \wedge C \quad \not\vdash A, B \wedge C \Rightarrow}{(A > B), (A \wedge B > C) \Rightarrow (A > B \wedge C)}\ \mathsf{c\_ag\_R}_2}{(A > B) \wedge (A \wedge B > C) \Rightarrow (A > B \wedge C)}\ \wedge_L$$

where the underivability premiss is the premiss of (RACT) and the other premisses are derivable using Lem. 6 since A, B, C are purely propositional. The case of (RAND) is similar, using $\mathsf{c\_ag\_CC}_1$. Deriving the axiomatic sequents is relatively straightforward using Lem. 6. □

Using cut admissibility we can finally show that the sequent systems indeed capture the corresponding I/O logics:

Theorem 12 (Equivalence). *Let $\mathcal{L}$ be one of the logics considered here. For every set $\{(A_1, X_1), \dots, (A_n, X_n)\}$ of I/O pairs we have*

$$\{(A_1, X_1), \dots, (A_n, X_n)\} \vdash_{\mathcal{L}} (A, X) \;\; \textit{iff} \\ \vdash_{\mathsf{G}_{\mathcal{L}}} (A_1 > X_1), \dots, (A_n > X_n) \Rightarrow (A > X)\ .$$

Proof. We use the fact that the construction of an I/O logic derivation corresponds to the construction of a sequent rules for $>$ using cuts.

The left to right direction is shown by induction on the depth of the I/O derivation, i.e., the maximal length of a branch in that derivation. We first consider an example of an I/O rule corresponding to an axiom from Fig. 4. Suppose that $\{(A_1, X_1), \dots, (A_n, X_n)\} \vdash_{\mathcal{L}} (A, X)$ and that the last applied rule was CT. Then there are $\mathcal{P}_1, \mathcal{P}_2$ with $\mathcal{P}_1 \cup \mathcal{P}_2 = \{(A_1, X_1), \dots, (A_n, X_n)\}$ and a formula Y such that $\mathcal{P}_1 \vdash_{\mathcal{L}} (A, Y)$ and $\mathcal{P}_2 \vdash_{\mathcal{L}} (A \wedge Y, X)$. Hence by induction hypothesis for the sets $\mathcal{P}_1^{>}$ and $\mathcal{P}_2^{>}$ of conditional formulae corresponding to the tuples in $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively we have $\vdash_{\mathsf{G}_{\mathcal{L}}} \mathcal{P}_1^{>} \Rightarrow (A > Y)$ and $\vdash_{\mathsf{G}_{\mathcal{L}}} \mathcal{P}_2^{>} \Rightarrow (A \wedge Y > X)$. By Lem. 11 we have $\vdash_{\mathsf{G}_{\mathcal{L}}} (A > Y) \wedge (A \wedge Y > X) \Rightarrow (A > X)$, and hence by invertibility of the propositional rules also $\vdash_{\mathsf{G}_{\mathcal{L}}} (A > Y), (A \wedge Y > X) \Rightarrow (A > X)$. Applying cut admissibility (Thm. 9) twice we have $\vdash_{\mathsf{G}_{\mathcal{L}}} \mathcal{P}_1^{>}, \mathcal{P}_2^{>} \Rightarrow (A > X)$, and admissibility of contraction (Lem. 8) yields the result. The cases for the other I/O rules corresponding to axiomatic sequents are similar.

As an example of an I/O rule corresponding to a rule from Fig. 4, assume that the last applied I/O rule was WO. Thus there is an I/O pair (A, Y) with $\{(A_1, X_1), \dots, (A_n, X_n)\} \vdash_{\mathcal{L}} (A, Y)$ and $Y \vdash X$. By induction hypothesis we have $\vdash_{\mathsf{G}_{\mathcal{L}}} (A_1 > X_1), \dots, (A_n > X_n) \Rightarrow (A > Y)$. Since $Y \vdash X$ propositionally and the propositional rules of $\mathsf{G}_{\mathcal{L}}$ are complete for classical propositional logic we also have $\vdash_{\mathsf{G}_{\mathcal{L}}} Y \Rightarrow X$, and Lem. 11 yields $\vdash_{\mathsf{G}_{\mathcal{L}}} (A > Y) \Rightarrow (A > X)$. Now admissibility of the cut rule gives the result.

For the right to left direction we use an induction on n, i.e., the number of I/O formulae on the left hand side of the sequent. For more details, see also Appendix A.2. If $\vdash_{\mathsf{G}_{\mathcal{L}}} (A_1 > X_1), \dots, (A_n > X_n) \Rightarrow (A > X)$, then the last applied rule must be a modal rule. Assume w.l.o.g. that all the $(A_i > X_i)$ are principal formulae (otherwise apply the induction hypothesis on the principal formulae). The base cases are those for $0 \leq n \leq 2$. In each case we distinguish subcases according to which rule was applied. We further use the fact that for propositional formulae we have $A \Rightarrow B$ iff $A \vdash B$.

Case $n = 0$: The last applied rule was one of $\mathsf{CC}_0, \mathsf{CCI}_0, \mathsf{R}_0, \mathsf{RI}_0$. For an application of CC_0 or R_0 with premiss $\Rightarrow D$ and conclusion $\Rightarrow (C > D)$ we first obtain $(\top, \top)$ from $\top$, which together with $C \vdash \top$ yields $(C, \top)$ by SI. This together with $\top \vdash D$ yields (C, D) by WO. The case of CCI_0 or RI_0 is even simpler, using ID and SI.

Case $n = 1$: The rules CC_1 and R_1 are straightforward using SI and WO. For the rule CCI_1 (and analogously for RI_1) we have:

$$\dfrac{C \Rightarrow A \quad B, C \Rightarrow D}{(A > B) \Rightarrow (C > D)}\ \mathsf{CCI}_1 \rightsquigarrow \dfrac{\dfrac{\dfrac{(A,B) \quad C \vdash A}{(C,B)}\ \mathsf{SI} \quad \dfrac{}{(C > C)}\ \mathsf{ID}}{(C, B \wedge C)}\ \mathsf{AND} \quad B \wedge C \vdash D}{(C,D)}\ \mathsf{WO}$$

For $\mathsf{ag\_CC}_1$ or $\mathsf{ag\_R}_1$ with premisses $C \Rightarrow A$ as well as $D \Rightarrow B$ and $B \Rightarrow D$ and conclusion $(A > B) \Rightarrow (C > D)$ we obtain (C, B) from (A, B) and $C \vdash A$ by SI. Together with $D \vdash B$ and $B \vdash D$ this yields (C, D) by OEQ. For $\mathsf{c\_ag\_CC}_1$ and $\mathsf{c\_ag\_R}_1$ we use the same derivation as for $\mathsf{ag\_CC}_1$. However, to ensure that we obtain a derivation valid in $\mathsf{c\_ag\_der}_1$ (resp. $\mathsf{c\_ag\_der}_3$) we need to check that none of the I/O pairs used as premisses is contradictory, i.e., specifically that $\nvdash A \wedge B \to \bot$. Assume that $\vdash A \wedge B \to \bot$. Then since $C \vdash A$ we also have $\vdash C \wedge B \to \bot$. Since moreover $D \vdash B$ and $B \vdash D$ we then obtain $\vdash C \wedge D \to \bot$, in contradiction to $\nvdash C, D \Rightarrow$. Thus $\nvdash A \wedge B \to \bot$.

Case $n = 2$: Rule CC_2 is straightforward using SI followed by AND and WO. For CCI_2 we also need to use ID and AND before WO. For R_2 we use SI followed by CT, AND and finally WO. For RI_2 we do the same but again insert ID and AND before the final application of WO. For $\mathsf{ag\_CC}_2$, suppose we have:

$$\dfrac{C \Rightarrow A_1 \quad C \Rightarrow A_2 \quad D \Rightarrow B_1 \quad D \Rightarrow B_2 \quad B_1, B_2 \Rightarrow D}{(A_1 > B_1), (A_2 > B_2) \Rightarrow (C > D)}\ \mathsf{ag\_CC}_2$$

Applying SI on (A_1, B_1) and $C \Rightarrow A_1$ yields (C, B_1) and similarly SI on (A_2, B_2) and $C \Rightarrow A_2$ yields (C, B_2). An application of AND then gives $(C, B_1 \wedge B_2)$, which together with $D \vdash B_1 \wedge B_2$ and $B_1 \wedge B_2 \vdash D$ by OEQ yields (C, D). The case of $\mathsf{ag\_R}_2$ is similar, using ACT instead of AND. For $\mathsf{c\_ag\_CC}_2$ we use the same derivation as for $\mathsf{ag\_CC}_2$. Additionally, we have to check that none of the I/O pairs occurring as assumptions of the derivation is inconsistent, i.e., that $\nvdash A_1 \wedge B_1 \to \bot$ and $\nvdash A_2 \wedge B_2 \to \bot$. From $\nvdash C, D \Rightarrow$ we obtain $\nvdash C \wedge D \to \bot$. Together with $\vdash C \leftrightarrow B_1 \wedge B_2$ this yields $\nvdash C \wedge B_1 \wedge B_2 \to \bot$. Since $C \vdash A_1$ and $C \vdash A_2$ this yields $\nvdash A_1 \wedge B_1 \wedge B_2 \to \bot$ and $\nvdash A_2 \wedge B_1 \wedge B_2 \to \bot$, and thus finally $\nvdash A_1 \wedge B_1 \to \bot$ and $\nvdash A_2 \wedge B_2 \to \bot$. The case of $\mathsf{c\_ag\_R}_2$ is analogous.

Case $n = m + 2$ with $m \geq 1$: We use essentially the method of proving soundness of "cuts between rules" from [11, Lem.2.4.5], using that the rules are constructed from smaller components via closure under cuts. I.e., for a rule with conclusion $(A_1 > B_1), \ldots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D)$ we construct a formula $(E > F)$ such that both $(A_1 > B_1), \ldots, (A_m > B_m), (E > F) \Rightarrow (C > D)$ and $(A_{m+1} > B_{m+1}), (A_{m+2} > B_{m+2}) \Rightarrow (E > F)$ are derivable given the original premisses. Then by induction hypothesis we obtain $\{(A_1, B_1), \ldots, (A_m, B_m), (E, F)\} \vdash_{\mathcal{L}} (C, D)$ and $\{(A_{m+1}, B_{m+1}), (A_{m+2}, B_{m+2})\} \vdash_{\mathcal{L}} (E, F)$ Putting these together we then have $\{(A_1, B_1), \ldots, (A_{m+2}, B_{m+2})\} \vdash_{\mathcal{L}} (C, D)$. For space reasons we only give the formula (E, F), assuming the rules as in Fig. 5 with conclusion $(A_1 > B_1), \ldots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D)$.

For CC_{m+2} we set $(E > F) = (C > B_{m+1} \wedge B_{m+2})$. For CCI_{m+2} we set $(E > F) = (C > C \wedge B_{m+1} \wedge B_{m+2})$. For R_{m+2} we use $(E > F) = (C \wedge \bigwedge_{i \leq m} B_i > B_{m+1} \wedge B_{m=2})$, and for RI_{m+2} we set $(E > F) = (C \wedge \bigwedge_{i \leq m} B_i > C \wedge \bigwedge_{i \leq m+2} B_i)$. For $\mathsf{ag\_CC}_{m+2}$ and $\mathsf{c\_ag\_CC}_{m+2}$ we use $(E > F) = (C > (B_{m+1} \wedge B_{m+2}) \vee D)$. In the case of $\mathsf{c\_ag\_CC}_{m+2}$ we additionally need to show the underivability premisses, i.e., $\nvdash C, (B_{m+2} \wedge B_{m+2}) \vee D \Rightarrow$. This follows by invertibility of the propositional rules from $\nvdash C, D \Rightarrow$. Finally, for $\mathsf{ag\_R}_{m+2}$ and $\mathsf{c\_ag\_R}_{m+2}$ we set $(E > F) = (C \wedge \bigwedge_{i \leq m} B_i > (B_{m+1} \wedge B_{m+2}) \vee D)$. In the case of $\mathsf{c\_ag\_R}_{m+2}$ again we also need to show the underivability premiss, i.e., $\nvdash C \wedge \bigwedge_{i \leq m} B_i, (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow$. Assume otherwise. Then by invertibility of the propositional rules we also have $\vdash C, B_1, \dots, B_{m+2} \Rightarrow$. Together with $\vdash D \Rightarrow B_i$ for $i \leq m+2$ and admissibility of cut and contraction this yields $\vdash C, D \Rightarrow$ in contradiction to the original premiss $\nvdash C, D \Rightarrow$. □

One benefit of the equivalence is that now we have a formal correspondence between certain I/O logics and the *conditional logics* or *dyadic deontic logics* obtained by adding (the Hilbert-style versions of) the axioms and rules of Fig. 4 to standard axioms for classical propositional logic:

Corollary 13. *For every finite set* $\{(A_i, X_i) : i \leq n\}$ *of I/O pairs we have* $\{(A_i, X_i) : i \leq n\} \vdash_{\mathcal{L}} (A, X)$ *iff* $\bigwedge_{i \leq n}(A_i > X_i) \to (A > X)$ *is a theorem of the conditional logic given by the corresponding axioms and rules of Fig. 4.*

Proof. The proof of Thm. 12 also shows that the sequent calculi are sound and complete for the logics given by the axioms and rules of Fig. 4 and the cut rule, and thus also the corresponding Hilbert-style systems. In particular, soundness is seen by converting the sequent rules into I/O derivations, then converting these into derivations using the axioms and rules of Fig. 4. □

This opens up new possibilities for comparing I/O logics to other conditional or dyadic deontic logics by investigating them in the same framework of Hilbert- or sequent systems, or by giving them semantics along the lines of [5]. As an example, Lewis' counterfactual logic $\mathbb{V}$ from [15] in the language with the dyadic *comparative plausibility* operator $\preccurlyeq$ has been equipped with a cut-free sequent system in [13]. The sequent rules for the operator $\preccurlyeq$ are given by the set $\{R_{n,m} : n \geq 1, m \geq 0\}$ for the rules

$$\frac{\{B_k \Rightarrow A_1, \dots, A_n, D_1, \dots, D_m : k \leq n\} \quad \{C_k \Rightarrow A_1, \dots, A_n, D_1, \dots, D_{k-1} : k \leq m\}}{\Gamma, (C_1 \preccurlyeq D_1), \dots, (C_m \preccurlyeq D_m) \Rightarrow (A_1 \preccurlyeq B_1), \dots, (A_n \preccurlyeq B_n), \Delta} \; R_{n,m}$$

Setting $n = 1$ we note that the structure of the premisses is the same as that of the rule RI_m, only with flipped right and left hand sides. Thus we obtain:

Theorem 14. *We have* $\{(A_1, X_1), \dots, (A_n, X_n)\} \vdash_{\mathsf{deriv}_3^+} (A, X)$ *if and only if* $\bigwedge_{i \leq n}(\neg A_i \preccurlyeq \neg X_i) \to (\neg A \preccurlyeq \neg X)$ *is a theorem of* $\mathbb{V}$.

Proof. Applications of the rule RI_m are simulated in the system for $\mathbb{V}$ by negation rules followed by $R_{1,n}$. Vice versa, if $(\neg A_1 \preccurlyeq \neg X_1), \dots, (\neg A_n \preccurlyeq X_n) \Rightarrow (\neg A \preccurlyeq$

$\neg X)$ is derivable in the system for $\mathbb{V}$, then w.l.o.g. it is the conclusion of an application of $\mathsf{R}_{1,n}$. Since the premisses are purely propositional, they are derivable in $\mathsf{G}_{\mathsf{deriv}_3^+}$. Using invertibility of the propositional rules we remove the negations, and then apply the rule RI_n. The full equivalence then follows from Thm. 12. □

Thus the I/O logic deriv_3^+ can be seen as the flat modal Horn fragment of conditional logic $\mathbb{V}$. This then yields an alternative semantics for deriv_3^+ in terms of the *sphere models* of [15] by simply spelling out the truth conditions for a formula $(\neg A \preccurlyeq \neg X)$.

4 Theorem Proving

One of the immediate benefits of the cut-free sequent calculi introduced above is that we immediately obtain an alternative decidability and complexity proof:

Theorem 15 (Decidability and Complexity). *Derivability in all the considered sequent systems is decidable in polynomial space.*

Proof. By a standard backwards proof search argument, e.g., the generic complexity result in [11, Thm. 2.7.8]. For the calculi with underivability statement we just need to flip the results for these statements. □

Since I/O pairs do not contain nested operators, the complexity for solving the entailment problem in I/O logics using our sequent calculi drops to the class Π_3^P of the polynomial hierarchy. However, since this is still above the optimal coNP-bounds following from [31] we do not consider this in detail here.

In terms of implementing our calculi, one suboptimal factor is that the number of principal formulae in the conclusion is unbounded, and that in contrast to, e.g., the rules for modal logic K the order of the principal formulae on the left hand side is crucial. We can obtain a more modular and arguably more elegant formulation by considering sequents with an additional *block*, in line with the idea of modularisation of sequent calculi in [14] and inspired by the blocks for nested sequent calculi in [1,19,8]. The main idea is to build up the sequent rules one formula at a time, starting with the principal formula on the right. The block is used to store the information during the building up of the rule.

Definition 16. *An* extended sequent *is a standard sequent possibly extended with a* block $[A > B : \Omega]$ *containing a formula* $A > B$ *and a multiset* Ω *of formulae. An extended sequent with a block is written* $\Gamma \Rightarrow \Delta, [A > B : \Omega]$.

The modal extended sequent rules then are given in Fig. 6, the modal part of the extended sequent calculi $\mathsf{EG}_{\mathcal{L}}$ is given in Fig. 7. In addition, all the calculi contain the standard propositional rules of Fig. 3. Note that this implies that the propositional rules can only be applied to standard sequents, i.e., sequents which do not contain a block. This is a design choice based purely on convenience, because it automatically separates the propositional and modal phases of a derivation, hence eliminating the need for a permutation-of-rules argument as

$$\frac{\Gamma \Rightarrow \Delta, [C > D :]}{\Gamma \Rightarrow \Delta, C > D} >_R \qquad \frac{\Omega \Rightarrow D}{\Gamma \Rightarrow \Delta, [C > D : \Omega]} \ \mathsf{jump} \qquad \frac{\Omega, C \Rightarrow D}{\Gamma \Rightarrow \Delta, [C > D : \Omega]} \ \mathsf{jump}^+$$

$$\frac{C \Rightarrow A \quad \Gamma \Rightarrow \Delta, [C > D : \Omega, B]}{\Gamma, A > B \Rightarrow \Delta, [C > D : \Omega]} >_L \qquad \frac{C, \Omega \Rightarrow A \quad \Gamma \Rightarrow \Delta, [C > D : \Omega, B]}{\Gamma, A > B \Rightarrow \Delta, [C > D : \Omega]} >_L^3$$

$$\frac{C \Rightarrow A \quad D \Rightarrow B \quad \Gamma \Rightarrow \Delta, [C > D : \Omega, B]}{\Gamma, A > B \Rightarrow \Delta, [C > D : \Omega]} >_L^{\mathsf{ag}} \qquad \frac{\Omega, B \Rightarrow D}{\Gamma \Rightarrow \Delta, [C > D : \Omega, B]} \ \mathsf{jump}^{\mathsf{ag}}$$

$$\frac{C, \Omega \Rightarrow A \quad D \Rightarrow B \quad \Gamma \Rightarrow \Delta, [C > D : \Omega, B]}{\Gamma, A > B \Rightarrow \Delta, [C > D : \Omega]} >_L^{\mathsf{ag\_3}} \qquad \frac{\Omega, B \Rightarrow D \quad \not\Rightarrow C, D \Rightarrow}{\Gamma \Rightarrow \Delta, [C > D : \Omega, B]} \ \mathsf{jump}^{\mathsf{c\_ag}}$$

Fig. 6. The extended sequent rules for internalised I/O logics

	$>_R$	$>_L$	$>_L^3$	$>_L^{\mathsf{ag}}$	$>_L^{\mathsf{ag\_3}}$	jump	jump$^+$	jump$^{\mathsf{ag}}$	jump$^{\mathsf{c\_ag}}$
$\mathsf{EG}_{\mathsf{deriv}_1}$	✓	✓				✓			
$\mathsf{EG}_{\mathsf{deriv}_1^+}$	✓	✓					✓		
$\mathsf{EG}_{\mathsf{deriv}_3}$	✓		✓			✓			
$\mathsf{EG}_{\mathsf{deriv}_3^+}$	✓		✓				✓		
$\mathsf{EG}_{\mathsf{ag\_der}_1}$	✓			✓				✓	
$\mathsf{EG}_{\mathsf{ag\_der}_3}$	✓				✓			✓	
$\mathsf{EG}_{\mathsf{c\_ag\_der}_1}$	✓			✓					✓
$\mathsf{EG}_{\mathsf{c\_ag\_der}_3}$	✓				✓				✓

Fig. 7. The exended sequent calculi

in [14, Thm.4.3] when showing equivalence of the calculi. Note also the subtle difference between the rules jump and jump$^{\mathsf{ag}}$: In the latter the left hand side of the premiss contains a formula B and hence cannot be empty. This ensures that the rule jump$^{\mathsf{ag}}$ can not be applied immediately above the rule $>_R$, capturing the fact that aggregative logics do not satisfy the axiom $\top$ of Fig. 4 and hence their sequent rules from Fig. 5 have a non-empty left hand side in the conclusion. The same mechanism holds for the consistent version jump$^{\mathsf{c\_ag}}$ of the rule.

Proposition 17. *The standard sequent calculi and the extended sequent calculi are equivalent, i.e., a standard sequent is derivable in* $\mathsf{G}_{\mathcal{L}}$ *if and only if it is derivable in* $\mathsf{EG}_{\mathcal{L}}$.

Proof. To see that every sequent derivable in the standard sequent calculi is also derivable in the corresponding extended sequent calculi it is enough to show that the standard modal rules are derivable rules in the extended sequent calculi. We do this by first applying (bottom-up) the right rule for $>$, followed by a number of applications of the left rule for $>$ and finally an application of the corresponding jump rule. E.g., an application

$$\frac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq n\} \quad B_1, \dots, B_n \Rightarrow D}{\Gamma, (A_1 > B_1), (A_2 > B_2), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta} \ \mathsf{R}_n$$

$$\dfrac{C, B_1, \ldots, B_{n-1} \Rightarrow A_n \qquad \dfrac{B_1, \ldots, B_n \Rightarrow D}{\varGamma \Rightarrow \varDelta, [C > D : B_1, \ldots, B_n]}\,\textsf{jump}}{\varGamma, (A_n > B_n) \Rightarrow \varDelta, [(C > D) : B_1, \ldots, B_{n-1}]}\,>_L^3$$

$$\vdots$$

$$\dfrac{\dfrac{C \Rightarrow A_1 \qquad \dfrac{C, B_1 \Rightarrow A_2 \qquad \varGamma, (A_3 > B_3), \ldots, (A_n > B_n) \Rightarrow \varDelta, [C > D : B_1, B_2]}{\varGamma, (A_2 > B_2), \ldots, (A_n > B_n) \Rightarrow \varDelta, [(C > D) : B_1]}\,>_L^3}{\varGamma, (A_1 > B_1), (A_2 > B_2), \ldots, (A_n > B_n) \Rightarrow \varDelta, [(C > D) :]}\,>_L^3}{\varGamma, (A_1 > B_1), (A_2 > B_2), \ldots, (A_n > B_n) \Rightarrow (C > D), \varDelta}\,>_R$$

Fig. 8. The derivation of the sequent rule R_n

of the rule R_n is simulated by the derivation in Fig. 8. The other cases are similar. For the other direction, due to the fact that an extended sequent contains at most one block and the shape of the rules, the modal rules are applied only in blocks with an application of $>_R$ at the bottom, followed by a number of applications of the appropriate version of the $>_L$ rule and finally a single application of the appropriate version of the $\textsf{jump}$ rule. Such blocks straightforwardly correspond to an application of the respective standard sequent rule, essentially reversing the simulation of that rule considered above. □

A prototype implementation of proof search in the extended sequent calculi in SWI-Prolog[1] is available as IOCondProver both as a web interface[2] and as source code on GitHub[3]. The implementation uses the Lean methodology [2] to delegate proof search to Prolog's backtracking mechanism. In case proof search is successful it outputs a LaTeX file containing the derivation, which is automatically rendered to a PDF file in the web interface.

While decision procedures for some I/O logics have been given using a semantic embedding into Higher Order Logic [3], it seems like the only other approach to automated reasoning for the logics considered here is that of the I/O Logics Workbench [29], which in its current version captures the logics deriv_1, deriv_1^+, deriv_3 and deriv_3^+ but does not seem to capture the logics $\mathsf{ag\_der}_1$, $\mathsf{ag\_der}_3$, $\mathsf{c\_ag\_der}_1$ or $\mathsf{c\_ag\_der}_3$. In contrast to the proof theoretic approach underlying IOCondProver, the reasoning underlying the I/O Logics Workbench is based on the semantic characterisation of the I/O logics, using a module for the consequence relation of an underlying base logic. While this makes the I/O Logics Workbench easily adaptable to other base logics such as intuitionistic logic, it also makes it difficult to adapt to the full nested logics characterised by the sequent calculi considered here. The proof theoretic approach of IOCondProver has the additional advantage of certificates for derivable sequents in the form of a derivation. Since IOCondProver is merely a prototype implementation, the focus of this article is on

[1] See `https://www.swi-prolog.org`
[2] See `http://subsell.logic.at/bprover/iocondprover/`
[3] See `https://github.com/blellmann/iocondprover`

the theoretical background, and in the absence of meaningful sets of benchmark formulae for I/O logics we do not consider a performance comparison with the I/O Logics Workbench here.

There are a number of calculi and theorem provers available both in the KLM-framework, see, e.g., [6,7,28] as well as in the framework of conditional logics, e.g., [9,18,20]. However, the vast majority of the available calculi and provers is based on KLM or conditional logics not corresponding to one of the I/O logics considered here. An exception is provided by the prover VINTE from [9], which implements proof search in an internal calculus for conditional logic $\mathbb{V}$. For the reasons given in the context of the I/O Logics Workbench we also do not consider performance comparisons with these provers here.

5 Conclusion

In this article we considered cut-free sequent calculi for a number of I/O logics including ones with consistency constraints. Two of the calculi are non-standard in that they contain underivability statements in the premisses. The calculi yield a correspondence of the original I/O logics to certain conditional logics and hence can form the basis of future comparisons between the two frameworks. We also considered modified versions of the calculi which are implemented in the prototype prover IOCondProver. For half of the considered logics this seems to be the first implementation available.

There are a number of possible directions for future research. The most obvious one is the extension to further I/O logics, in particular the logics deriv_2 and deriv_4, which result from deriv_1 and deriv_3 by adding essentially the axiom $(A > C) \land (B > C) \to (A \lor B > C)$. Since all the axioms for these logics have modal Horn form a cut-free sequent system immediately follows from the generic construction of [11, Sec. 4.1, Cor. 4.1.20]. The rules could even be made comprehensible by using the universal orders of derivation of [16, Sec. 8], since the order of applying I/O rules corresponds to an order in the construction of the sequent rules by cuts. Unfortunately Contraction might not be admissible in the resulting systems, and hence it is not clear that they can be used for automated reasoning. In contrast, it might be simpler to adapt the calculi considered here to the operators for *permissions* considered in [17,21]. Since the logics considered here are given by axioms in modal Horn form and do not involve disjunction it might also be straightforward to adapt them to intuitionistic instead of classical logic as the base logic in the spirit of [22]. Finally, it might be possible to exploit the sequent formulation in order to give a constructive proof of an analog of the Craig Interpolation Property for I/O logics, following the methods of [13,11].

Acknowledgements. This article would not have been possible without the many discussions on the topic with Leon van der Torre. I also thank the reviewers for their thorough reading and comments which helped to improve the article.

References

1. Alenda, R., Olivetti, N., Pozzato, G.L.: Nested sequent calculi for normal conditional logics. J. Log. Comput. **26**(1), 7–50 (2013). https://doi.org/10.1093/logcom/ext034
2. Becker, B., Posegga, J.: leanTAP: Lean tableau-based deduction. Journal of Automated Reasoning **15**, 339–358 (1995)
3. Benzmüller, C., Farjami, A., Meder, P., Parent, X.: I/O logics in HOL. Journal of Applied Logics **6**(5), 715–754 (2019)
4. Bonatti, P.A., Olivetti, N.: Sequent calculi for propositional nonmonotonic logics. ACM Transactions on Computational Logic **3**, 226–278 (2002)
5. Chellas, B.F.: Modal Logic. Cambridge University Press (1980)
6. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: Analytic tableaux calculi for KLM logics of nonmonotonic reasoning. ACM Trans. Comput. Log. **10**(3) (2009)
7. Giordano, L., Gliozzi, V., Pozzato, G.L.: KLMLean 2.0: A theorem prover for KLM logics of nonmonotonic reasoning. In: Olivetti, N. (ed.) TABLEAUX 2007, LNAI, vol. 4548, pp. 238–244. Springer (2007)
8. Girlando, M., Lellmann, B., Olivetti, N., Pozzato, G.L.: Standard sequent calculi for Lewis' logics of counterfactuals. In: Michael, L., Kakas, A. (eds.) Logics in Artificial Intelligence. JELIA 2016., Lecture Notes in Computer Science, vol. 10021, pp. 272–287. Springer, Cham (2016)
9. Girlando, M., Lellmann, B., Olivetti, N., Pozzato, G.L., Vitalis, Q.: VINTE: An implementation of internal calculi for lewis' logics of counterfactual reasoning. In: Schmidt, R.A., Nalon, C. (eds.) TABLEAUX 2017, LNCS, vol. 10501, pp. 149–159. Springer (2017)
10. Kraus, S., Lehman, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence **44**(1-2), 167–207 (1990)
11. Lellmann, B.: Sequent Calculi with Context Restrictions and Applications to Conditional Logic. Ph.D. thesis, Imperial College London (2013), `http://hdl.handle.net/10044/1/18059`
12. Lellmann, B., Pattinson, D.: Cut elimination for shallow modal logics. In: Brünnler, K., Metcalfe, G. (eds.) TABLEAUX 2011, LNAI, vol. 6793, pp. 211–225. Springer-Verlag Berlin Heidelberg (2011)
13. Lellmann, B., Pattinson, D.: Sequent systems for Lewis' conditional logics. In: del Cerro, L.F.n., Herzig, A., Mengin, J. (eds.) JELIA 2012, LNAI, vol. 7519, pp. 320–332. Springer-Verlag Berlin Heidelberg (2012)
14. Lellmann, B., Pimentel, E.: Modularisation of sequent calculi for normal and non-normal modalities. ACM Trans. Comput. Logic **20**(2), 7:1–7:46 (2019)
15. Lewis, D.: Counterfactuals. Blackwell (1973)
16. Makinson, D., van der Torre, L.: Input/output logics. Journal of Philosophical Logic **29**, 383–408 (2000)
17. Makinson, D., van der Torre, L.: Permission from an input/output perspective. Journal of Philosophical Logic **32**, 391–416 (2003)
18. Olivetti, N., Pozzato, G.L.: Condlean 3.0: Improving condlean for stronger conditional logics. In: TABLEAUX. pp. 328—332 (2005), `http://www.springerlink.com/index/835jd3u9fx8klwy9.pdf`
19. Olivetti, N., Pozzato, G.L.: A standard internal calculus for Lewis' counterfactual logics. In: Nivelle, H.D. (ed.) TABLEAUX 2015, Lecture Notes in Artificial Intelligence, vol. 9323, pp. 270–286. Springer International Publishing (2015)
20. Olivetti, N., Pozzato, G.: Nescond: An implementation of nested sequent calculi for conditional logics. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) IJCAR 2014,

Lecture Notes in Computer Science, vol. 8562, pp. 511–518. Springer International Publishing (2014)
21. Olszewski, M., Parent, X., van der Torre, L.: Input/output logic with a consistency check – the case of permission. In: DEON 2020/2021. College Publications (2021)
22. Parent, X., Gabbay, D., van der Torre, L.: Intuitionistic basis for input/output logic. In: Hansson, S. (ed.) David Makinson on Classical Methods for Non-Classical Problems, pp. 263–286. Outstanding Contributions to Logic, Springer (2014)
23. Parent, X., van der Torre, L.: Input/output logic. In: Gabbay, D., Horty, J., Parent, X., van der Meyden, R., van der Torre, L. (eds.) Handbook of Deontic Logic and Normative Systems, chap. 8, pp. 495–544. College Publications (2013)
24. Parent, X., van der Torre, L.: The pragmatic oddity in norm-based deontic logics. In: Governatori, G. (ed.) ICAIL 2017, pp. 169–178. Association for Computing Machinery (2017). https://doi.org/http://dx.doi.org/10.1145/3086512.3086529
25. Parent, X., van der Torre, L.: I/O logics with a consistency check. In: Broersen, J., Condoravdi, C., Nair, S., Pigozzi, G. (eds.) Deontic Logic and Normative Systems. DEON 2018, pp. 285–300. College Publications (2018)
26. Parent, X., van der Torre, L.: Input/output logics without weakening. Filosofiska Notiser **6**(1), 189–208 (2019)
27. Pattinson, D., Schröder, L.: Generic modal cut elimination applied to conditional logics. Log. Methods Comput. Sci. **7**(1:4), 1–28 (2011)
28. Pozzato, G.L.: Conditional and Preferential Logics: Proof Methods and Theorem Proving, Frontiers in Artificial Intelligence and Applications, vol. 208. IOS Press Amsterdam (2010)
29. Steen, A.: Goal-directed decision procedures for input/output logics. In: Liu, F., Marra, A., Portner, P., Putte, F.V.D. (eds.) DEON2020/2021. College Publications (2021 (to appear))
30. Stolpe, A.: Normative consequence: The problem of keeping it whilst giving it up. In: van der Meyden, R., van der Torre, L. (eds.) DEON 2008, LNAI, vol. 5076, pp. 174–188. Springer (2008)
31. Sun, X., Robaldo, L.: On the complexity of input/output logic. Journal of Applied Logic (2017)
32. Troelstra, A.S., Schwichtenberg, H.: Basic Proof Theory, Cambridge Tracts In Theoretical Computer Science, vol. 43. Cambridge University Press, 2 edn. (2000)

A Appendix

A.1 Additional Details for the Proof of Theorem 9 (Cut Elimination)

The full list of cases for the modal rules is as follows, distinguishing which logic we are looking at:

Subcase $\mathcal{L} = \mathsf{deriv}_1$: If the last rules were CC_n and CC_m we have the conclusion

$$\frac{\{C \Rightarrow A_i : i \leq n\} \quad B_1, \ldots, B_n \Rightarrow D}{(A_1 > B_1), \ldots, (A_n > B_n) \Rightarrow (C > D)} \; \mathsf{CC}_n$$

and

$$\frac{\{G \Rightarrow E_i : k \neq i \leq m\} \quad G \Rightarrow C \quad F_1, \ldots, F_{k-1}, D, F_{k+1}, \ldots, F_m \Rightarrow H}{(E_1 > F_1), \ldots, (E_{k-1} > F_{k-1}), (C > D), (E_{k+1} > F_{k+1}), \ldots, (E_m > F_m) \Rightarrow (G > H)} \; \mathsf{CC}_m$$

Applying cuts on smaller formulae to the premisses yields:

$$\{G \Rightarrow E_i : k \neq i \leq m\} \quad \{G \Rightarrow A_i : i \leq n\} \quad F_1, \ldots, F_{k-1}, B_1, \ldots, B_n, F_{k+1}, \ldots, F_m \Rightarrow H$$

Now an application of CC_{n+m-1} yields the result

$$(E_1 > F_1), \ldots, (E_{k-1} > F_{k-1}), (A_1 > B_1), \ldots, (A_n > B_n), (E_{k+1} > F_{k+1}), \ldots, (E_m > F_m) \Rightarrow (G > H)$$

Subcase $\mathcal{L} = \mathsf{deriv}_1^+$: If the last rules were CCI_n and CCI_m we have

$$\frac{\{C \Rightarrow A_i : i \leq n\} \quad C, B_1, \ldots, B_n \Rightarrow D}{(A_1 > B_1), \ldots, (A_n > B_n) \Rightarrow (C > D)} \; \mathsf{CCI}_n$$

and

$$\frac{\{G \Rightarrow E_i : k \neq i \leq m\} \quad G \Rightarrow C \quad G, F_1, \ldots, F_{k-1}, D, F_{k+1}, \ldots, F_m \Rightarrow H}{(E_1 > F_1), \ldots, (E_{k-1} > F_{k-1}), (C > D), (E_{k+1} > F_{k+1}), \ldots, (E_m > F_m) \Rightarrow (G > H)} \; \mathsf{CCI}_m$$

Applying cuts on smaller formulae to the premisses yields:

$$\begin{array}{l} \{G \Rightarrow E_i : k \neq i \leq m\} \\ \quad \{G \Rightarrow A_i : i \leq n\} \\ \qquad G, F_1, \ldots, F_{k-1}, B_1, \ldots, B_n, F_{k+1}, \ldots, F_m \Rightarrow H \end{array}$$

Now an application of CCI_{n+m-1} yields the result

$$(E_1 > F_1), \ldots, (E_{k-1} > F_{k-1}), (A_1 > B_1), \ldots, (A_n > B_n), (E_{k+1} > F_{k+1}), \ldots, (E_m > F_m) \Rightarrow (G > H)$$

Subcase $\mathcal{L} = \mathsf{deriv}_3$: Suppose we have an application

$$\frac{\{C, B_1, \ldots, B_{i-1} \Rightarrow A_i : i \leq n\} \quad B_1, \ldots, B_n \Rightarrow D}{(A_1 > B_1), \ldots, (A_n > B_n) \Rightarrow (C > D)} \; \mathsf{R}_n$$

and

$$\dfrac{\begin{array}{l}\{G, F_1, \dots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad G, F_1, \dots, F_{k-1} \Rightarrow C \\ \qquad \{G, F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \qquad\quad F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_m \Rightarrow H\end{array}}{(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (C > D), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)} \; \mathsf{R}_m$$

Applying cut of smaller complexity to the premisses yields

$$\begin{array}{l}\{G, F_1, \dots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad \{G, F_1, \dots, F_{k-1}, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq n\} \\ \qquad \{G, F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \qquad\quad F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_m \Rightarrow H\end{array}$$

and an application of R_{n+m-1} yields the result:

$$(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (A_1 > B_1), \dots, (A_n > B_n), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)$$

Subcase $\mathcal{L} = \mathsf{deriv}_3^+$: Suppose we have an application

$$\dfrac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq n\} \quad C, B_1, \dots, B_n \Rightarrow D}{(A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D)} \; \mathsf{RI}_n$$

and

$$\dfrac{\begin{array}{l}\{G, F_1, \dots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad G, F_1, \dots, F_{k-1} \Rightarrow C \\ \qquad \{G, F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \qquad\quad G, F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_m \Rightarrow H\end{array}}{(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (C > D), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)} \; \mathsf{RI}_m$$

Applying cut of smaller complexity to the premisses yields

$$\begin{array}{l}\{G, F_1, \dots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad \{G, F_1, \dots, F_{k-1}, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq n\} \\ \qquad \{G, F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \qquad\quad G, F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_m \Rightarrow H\end{array}$$

and an application of RI_{n+m-1} yields the result:

$$(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (A_1 > B_1), \dots, (A_n > B_n), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)$$

Subcase $\mathcal{L} = \mathsf{ag\_der}_1$: Suppose we have

$$\dfrac{\{C \Rightarrow A_i : 1 \leq i \leq n\} \quad \{D \Rightarrow B_i : 1 \leq i \leq n\} \quad B_1, \dots, B_n \Rightarrow D}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta} \; \mathsf{ag\_CC}_n$$

and

$$\dfrac{\begin{array}{l}\{G \Rightarrow E_i : k \neq i \leq m\} \\ \quad G \Rightarrow C \\ \quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad H \Rightarrow D \\ \quad\quad\quad\quad F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_m \Rightarrow H\end{array}}{(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (C > D), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)}\ \mathsf{ag\_CC}_m$$

Cuts on formulae of smaller complexity in the premisses yield

$$\begin{array}{l}\{G \Rightarrow E_i : k \neq i \leq m\} \\ \quad \{G \Rightarrow A_i : i \leq n\} \\ \quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad \{H \Rightarrow B_i : i \leq n\} \\ \quad\quad\quad\quad F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_m \Rightarrow H\end{array}$$

and an application of $\mathsf{ag\_CC}_{n+m-1}$ yields the result.

$$(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (A_1 > B_1), \dots, (A_n > B_n), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)$$

Subcase $\mathcal{L} = \mathsf{ag\_der}_3$: Suppose we have an application

$$\dfrac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq n\} \quad \{D \Rightarrow B_i : i \leq n\} \quad B_1, \dots, B_n \Rightarrow D}{(A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D)}\ \mathsf{ag\_R}_n$$

and

$$\dfrac{\begin{array}{l}\{G, F_1, \dots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad G, F_1, \dots, F_{k-1} \Rightarrow C \\ \quad\quad \{G, F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \quad\quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad\quad H \Rightarrow D \\ \quad\quad\quad\quad\quad F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_m \Rightarrow H\end{array}}{(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (C > D), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)}\ \mathsf{ag\_R}_m$$

Applying cut of smaller complexity to the premisses yields

$$\begin{array}{l}\{G, F_1, \dots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad \{G, F_1, \dots, F_{k-1}, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq n\} \\ \quad\quad \{G, F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \quad\quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad\quad \{H \Rightarrow B_i : i \leq n\} \\ \quad\quad\quad\quad\quad F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_m \Rightarrow H\end{array}$$

and an application of $\mathsf{ag\_R}_{n+m-1}$ yields the result:

$$(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (A_1 > B_1), \dots, (A_n > B_n), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)$$

Subcase $\mathcal{L} = \mathsf{c\_ag\_der}_1$: Suppose we have

$$\frac{\{C \Rightarrow A_i : 1 \leq i \leq n\} \quad \{D \Rightarrow B_i : 1 \leq i \leq n\} \quad B_1, \dots, B_n \Rightarrow D \quad \nvdash C, D \Rightarrow}{\Gamma, (A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D), \Delta}\ \mathsf{c\_ag\_CC}_n$$

and

$$\frac{\begin{array}{l}\{G \Rightarrow E_i : k \neq i \leq m\} \\ \quad G \Rightarrow C \\ \quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad H \Rightarrow D \\ \quad\quad\quad\quad F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_m \Rightarrow H \\ \quad\quad\quad\quad\quad \nvdash G, H \Rightarrow \end{array}}{(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (C > D), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)}\ \mathsf{c\_ag\_CC}_m$$

Cuts on formulae of smaller complexity in the premisses yield

$$\begin{array}{l}\{G \Rightarrow E_i : k \neq i \leq m\} \\ \quad \{G \Rightarrow A_i : i \leq n\} \\ \quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad \{H \Rightarrow B_i : i \leq n\} \\ \quad\quad\quad\quad F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_m \Rightarrow H \\ \quad\quad\quad\quad\quad \nvdash G, H \Rightarrow \end{array}$$

and an application of $\mathsf{c\_ag\_CC}_{n+m-1}$ yields the result:

$$(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (A_1 > B_1), \dots, (A_n > B_n), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)$$

Subcase $\mathcal{L} = \mathsf{c\_ag\_der}_3$: Suppose we have an application

$$\frac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq n\} \quad \{D \Rightarrow B_i : i \leq n\} \quad B_1, \dots, B_n \Rightarrow D \quad \nvdash C, D \Rightarrow}{(A_1 > B_1), \dots, (A_n > B_n) \Rightarrow (C > D)}\ \mathsf{c\_ag\_R}_n$$

and

$$\frac{\begin{array}{l}\{G, F_1, \dots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad G, F_1, \dots, F_{k-1} \Rightarrow C \\ \quad\quad \{G, F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \quad\quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad\quad H \Rightarrow D \\ \quad\quad\quad\quad\quad F_1, \dots, F_{k-1}, D, F_{k+1}, \dots, F_m \Rightarrow H \\ \quad\quad\quad\quad\quad\quad \nvdash G, H \Rightarrow \end{array}}{(E_1 > F_1), \dots, (E_{k-1} > F_{k-1}), (C > D), (E_{k+1} > F_{k+1}), \dots, (E_m > F_m) \Rightarrow (G > H)}\ \mathsf{c\_ag\_R}_m$$

Applying cut of smaller complexity to the premisses yields

$$\begin{array}{l}\{G, F_1, \dots, F_{i-1} \Rightarrow E_i : i \leq k-1\} \\ \quad \{G, F_1, \dots, F_{k-1}, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq n\} \\ \quad\quad \{G, F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_{i-1} \Rightarrow E_i : k < i \leq m\} \\ \quad\quad\quad \{H \Rightarrow F_i : k \neq i \leq m\} \\ \quad\quad\quad\quad \{H \Rightarrow B_i : i \leq n\} \\ \quad\quad\quad\quad\quad F_1, \dots, F_{k-1}, B_1, \dots, B_n, F_{k+1}, \dots, F_m \Rightarrow H \\ \quad\quad\quad\quad\quad\quad \nvdash G, H \Rightarrow \end{array}$$

and an application of $\mathsf{c\_ag\_R}_{n+m-1}$ yields the result:

$$(E_1 > F_1), \ldots, (E_{k-1} > F_{k-1}), (A_1 > B_1), \ldots, (A_n > B_n), (E_{k+1} > F_{k+1}), \ldots, (E_m > F_m) \Rightarrow (G > H)$$

A.2 Additional Details for the Proof of Thm. 12 (Equivalence with the I/O systems)

The full list of cases for the right to left direction is as follows.

Case $n = 0$: The last applied rule was one of $\mathsf{CC}_0, \mathsf{CCI}_0, \mathsf{R}_0, \mathsf{RI}_0$.
Rule applied: CC_0:

$$\dfrac{\Rightarrow D}{\Rightarrow (C > D)}\,\mathsf{CC}_0 \quad\rightsquigarrow\quad \dfrac{\dfrac{\dfrac{}{(\top,\top)}\,\top \quad C \vdash \top}{(C,\top)}\,\mathsf{SI} \quad \top \vdash D}{(C,D)}\,\mathsf{WO}$$

Rule applied: R_0: Same as for CC_0.
Rule applied: CCI_0:

$$\dfrac{C \Rightarrow D}{\Rightarrow (C > D)}\,\mathsf{CCI}_0 \rightsquigarrow \dfrac{\dfrac{}{(D,D)}\,\mathsf{ID} \quad C \vdash D}{(C,D)}\,\mathsf{SI}$$

Rule applied: RI_0: Same as for CCI_0.

Case $n = 1$: Rule applied: CC_1:

$$\dfrac{C \Rightarrow A \quad B \Rightarrow D}{(A > B) \Rightarrow (C > D)}\,\mathsf{CC}_1 \rightsquigarrow \dfrac{\dfrac{(A,B) \quad C \vdash A}{(C,B)}\,\mathsf{SI} \quad B \vdash D}{(C,D)}\,\mathsf{WO}$$

Subcase CCI_1:

$$\dfrac{C \Rightarrow A \quad B, C \Rightarrow D}{(A > B) \Rightarrow (C > D)}\,\mathsf{CCI}_1 \rightsquigarrow \dfrac{\dfrac{\dfrac{(A,B) \quad C \vdash A}{(C,B)}\,\mathsf{SI} \quad \dfrac{}{(C > C)}\,\mathsf{ID}}{(C, B \wedge C)}\,\mathsf{AND} \quad B \wedge C \vdash D}{(C,D)}\,\mathsf{WO}$$

Subcase R_1: Same as CC_1.
Subcase RI_1: Same as CCI_1.
Subcase $\mathsf{ag\_CC}_1$:

$$\dfrac{C \Rightarrow A \quad D \Rightarrow B \quad B \Rightarrow D}{(A > B) \Rightarrow (C > D)}\,\mathsf{ag\_CC}_1 \rightsquigarrow \dfrac{\dfrac{(A,B) \quad C \vdash A}{(C,B)}\,\mathsf{SI} \quad D \vdash B \quad B \vdash D}{(C,D)}\,\mathsf{OEQ}$$

Subcase $\mathsf{ag\_R}_1$: Same as for $\mathsf{ag\_CC}_1$.

Subcase $\mathsf{c\_ag\_CC}_1$: We use the same derivation as in the case of $\mathsf{ag\_CC}_1$. However, to ensure that we obtain a derivation valid in $\mathsf{c\_ag\_der}_1$ we need to check that none of the I/O pairs used as premisses is contradictory, i.e., specifically that $\nvdash A \land B \to \bot$. Assume that $\vdash A \land B \to \bot$. Then since $C \vdash A$ we also have $\vdash C \land B \to \bot$. Since moreover $D \vdash B$ and $B \vdash D$ we then obtain $\vdash C \land D \to \bot$, in contradiction to $\nvdash C, D \Rightarrow\ $. Thus $\nvdash A \land B \to \bot$.

Subcase $\mathsf{c\_ag\_R}_1$: Same as for $\mathsf{c\_ag\_CC}_1$.

Case $n = 2$: Subcase CC_2:

$$\dfrac{C \Rightarrow A_1 \quad C \Rightarrow A_2 \quad B_1, B_2 \Rightarrow D}{(A_1 > B_1), (A_2 > B_2) \Rightarrow (C > D)}\ \mathsf{CC}_2$$

$$\rightsquigarrow \quad \dfrac{\dfrac{\dfrac{(A_1, B_1) \quad C \vdash A_1}{(C, B_1)}\ \mathsf{SI} \qquad \dfrac{(A_2, B_2) \quad C \vdash A_2}{(C, B_2)}\ \mathsf{SI}}{(C, B_1 \land B_2)}\ \mathsf{AND} \qquad B_1 \land B_2 \vdash D}{(C, D)}\ \mathsf{WO}$$

Subcase CCI_2:

$$\dfrac{C \Rightarrow A_1 \quad C \Rightarrow A_2 \quad B_1, B_2, C \Rightarrow D}{(A_1 > B_1), (A_2 > B_2) \Rightarrow (C > D)}$$

$$\rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\dfrac{(A_1, B_1) \quad C \vdash A_1}{(C, B_1)}\ \mathsf{SI} \qquad \dfrac{(A_2, B_2) \quad C \vdash A_2}{(C, B_2)}\ \mathsf{SI}}{(C, B_1 \land B_2)}\ \mathsf{AND} \qquad \dfrac{}{(C, C)}\ \mathsf{ID}}{(C, B_1 \land B_2 \land C)}\ \mathsf{AND} \qquad B_1 \land B_2 \land C \vdash D}{(C, D)}\ \mathsf{WO}$$

Subcase R_2:

$$\dfrac{C \Rightarrow A_1 \quad C, B_1 \Rightarrow A_2 \quad B_1, B_2 \Rightarrow D}{(A_1 > B_1), (A_2 > B_2) \Rightarrow (C > D)}\ \mathsf{R}_2$$

$$\rightsquigarrow \quad \dfrac{\dfrac{\dfrac{(A_1, B_1) \quad C \vdash A_1}{(C, B_1)}\ \mathsf{SI} \qquad \dfrac{\dfrac{(A_1, B_1) \quad C \vdash A_1}{(C, B_1)}\ \mathsf{SI} \qquad \dfrac{(A_2, B_2) \quad C \land B_1 \vdash A_2}{(C \land B_1, B_2)}\ \mathsf{SI}}{(C, B_2)}\ \mathsf{CT}}{(C, B_1 \land B_2)}\ \mathsf{AND} \qquad B_1 \land B_2 \vdash D}{(C, D)}\ \mathsf{WO}$$

Subcase RI_2:

$$\dfrac{C \Rightarrow A_1 \quad C, B_1 \Rightarrow A_2 \quad C, B_1, B_2 \Rightarrow D}{(A_1 > B_1), (A_2 > B_2) \Rightarrow (C > D)}$$

$$\rightsquigarrow \quad \dfrac{\dfrac{\begin{array}{c}(A_1, B_1) \quad C \vdash A_1 \quad (A_2, B_2) \quad C \land B_1 \vdash A_2 \\ \vdots\ \mathcal{D} \\ (C, B_1 \land B_2)\end{array} \qquad \dfrac{}{(C, C)}\ \mathsf{ID}}{(C, C \land B_1 \land B_2)}\ \mathsf{AND} \qquad C \land B_1 \land B_2 \vdash D}{(C, D)}\ \mathsf{WO}$$

where $\mathcal{D}$ is the subderivation from the case R_2.

Subcase $\mathsf{ag\_CC}_2$:

$$\frac{C \Rightarrow A_1 \quad C \Rightarrow A_2 \quad D \Rightarrow B_1 \quad D \Rightarrow B_2 \quad B_1, B_2 \Rightarrow D}{(A_1 > B_1), (A_2 > B_2) \Rightarrow (C > D)}\ \mathsf{ag\_CC}_2$$

$$\dfrac{\dfrac{\dfrac{(A_1, B_1) \quad C \vdash A_1}{(C, B_1)}\ \mathsf{SI} \quad \dfrac{(A_2, B_2) \quad C \vdash A_2}{(C, B_2)}\ \mathsf{SI}}{(C, B_1 \wedge B_2)}\ \mathsf{AND} \quad D \vdash B_1 \wedge B_2 \quad B_1 \wedge B_2 \vdash D}{(C, D)}\ \mathsf{OEQ}$$

Subcase $\mathsf{ag\_R}_2$:

$$\frac{C \Rightarrow A_1 \quad C, B_1 \Rightarrow A_2 \quad D \Rightarrow B_1 \quad D \Rightarrow B_2 \quad B_1, B_2 \Rightarrow D}{(A_1 > B_1), (A_2 > B_2) \Rightarrow (C > D)}\ \mathsf{ag\_R}_2$$

$$\rightsquigarrow \quad \dfrac{\dfrac{\dfrac{(A_1, B_1) \quad C \vdash A_1}{(C, B_1)}\ \mathsf{SI} \quad \dfrac{(A_2, B_2) \quad C \wedge B_1 \vdash A_2}{(C \wedge B_1, B_2)}\ \mathsf{SI}}{(C, B_1 \wedge B_2)}\ \mathsf{ACT} \quad D \vdash B_1 \wedge B_2 \quad B_1 \wedge B_2 \vdash D}{(C, D)}\ \mathsf{OEQ}$$

Subcase $\mathsf{c\_ag\_CC}_2$:

$$\frac{C \Rightarrow A_1 \quad C \Rightarrow A_2 \quad D \Rightarrow B_1 \quad D \Rightarrow B_2 \quad B_1, B_2 \Rightarrow D \quad \nvdash C, D \Rightarrow}{(A_1 > B_1), (A_2 > B_2) \Rightarrow (C > D)}\ \mathsf{ag\_CC}_2$$

Again, we use the derivation from subcase $\mathsf{ag\_CC}_2$. We have to check that none of the I/O pairs occurring as assumptions of the derivation is inconsistent, i.e., that $\nvdash A_1 \wedge B_1 \to \bot$ and $\nvdash A_2 \wedge B_2 \to \bot$. From $\nvdash C, D \Rightarrow$ we obtain $\nvdash C \wedge D \to \bot$. Together with $\vdash C \leftrightarrow B_1 \wedge B_2$ this yields $\nvdash C \wedge B_1 \wedge B_2 \to \bot$. Since $C \vdash A_1$ and $C \vdash A_2$ this further yields $\nvdash A_1 \wedge B_1 \wedge B_2 \to \bot$ and $\nvdash A_2 \wedge B_1 \wedge B_2 \to \bot$, and thus finally $\nvdash A_1 \wedge B_1 \to \bot$ and $\nvdash A_2 \wedge B_2 \to \bot$.

Subcase $\mathsf{c\_ag\_R}_2$:

$$\frac{C \Rightarrow A_1 \quad C, B_1 \Rightarrow A_2 \quad D \Rightarrow B_1 \quad D \Rightarrow B_2 \quad B_1, B_2 \Rightarrow D \quad \nvdash C, D \Rightarrow}{(A_1 > B_1), (A_2 > B_2) \Rightarrow (C > D)}\ \mathsf{ag\_R}_2$$

Analogous to the previous case we use the derivation from subcase $\mathsf{ag\_R}_2$ and the reasoning as in the subcase $\mathsf{c\_ag\_CC}_2$ to obtain $\nvdash A_1 \wedge B_1 \to \bot$ and $\nvdash A_2 \wedge B_2 \to \bot$.

Case $n = m + 2$ with $m \geq 1$: We use essentially the method of proving soundness of "cuts between rules" from [11, Lem.2.4.5], using that the rules are constructed from smaller components via closure under cuts.

Subcase CC_{m+2}:

$$\frac{\{C \Rightarrow A_i : i \leq m+2\} \quad B_1, \dots, B_{m+2} \Rightarrow D}{(A_1 > B_1), \dots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D)}\ \mathsf{CC}_{m+2}$$

We deconstruct the rule CC_{m+2} using the formula $(C > B_{m+1} \wedge B_{m+2})$ into the following two rules:

$$\frac{\{C \Rightarrow A_i : i \leq m\} \quad C \Rightarrow C \quad B_1, \dots, B_m, B_{m+1} \wedge B_{m+2} \Rightarrow D}{(A_1 > B_1), \dots, (A_m > B_m), (C > B_{m+1} \wedge B_{m+2}) \Rightarrow (C > D)} \ \mathsf{CC}_{m+1}$$

$$\frac{C \Rightarrow A_{m+1} \quad C \Rightarrow A_{m+2} \quad B_{m+1}, B_{m+2} \Rightarrow B_{m+1} \wedge B_{m+2}}{(A_{m+1} > B_{m+1}), (A_{m+2} > B_{m+2}) \Rightarrow (C > B_{m+1} \wedge B_{m+2})} \ \mathsf{CC}_2$$

Since the premisses of these rules are derivable from the premisses of the original rule, their conclusions are derivable. Hence we can use the induction hypothesis to obtain

$$\{(A_i, B_i) : i \leq m\} \cup \{(C, B_{m+1} \wedge B_{m+2})\} \vdash_{\mathcal{L}} (C, D)$$

and

$$\{(A_{m+1}, B_{m+1}), (A_{m+2}, B_{m+2})\} \vdash_{\mathcal{L}} (C, B_{m+1} \wedge B_{m+2})$$

Piecing these together we obtain $\{(A_i, B_i) : i \leq m+2\} \vdash_{\mathcal{L}} (C, D)$.

Subcase CCI_{m+2}:

$$\frac{\{C \Rightarrow A_i : i \leq m+2\} \quad C, B_1, \dots, B_{m+2} \Rightarrow D}{(A_1 > B_1), \dots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D)} \ \mathsf{CCI}_{m+2}$$

is deconstructed using the formula $(C > C \wedge B_{m+1} \wedge B_{m+2})$ into

$$\frac{\{C \Rightarrow A_i : i \leq m\} \quad C \Rightarrow C \quad C, B_1, \dots, B_m, C \wedge B_{m+1} \wedge B_{m+2} \Rightarrow D}{(A_1 > B_1), \dots, (A_m > B_m), (C > C \wedge B_{m+1} \wedge B_{m+2}) \Rightarrow (C > D)} \ \mathsf{CCI}_{m+1}$$

$$\frac{C \Rightarrow A_{m+1} \quad C \Rightarrow A_{m+2} \quad C, B_{m+1}, B_{m+2} \Rightarrow C \wedge B_{m+1} \wedge B_{m+2}}{(A_{m+1} > B_{m+1}), (A_{m+2} > B_{m+2}) \Rightarrow (C > C \wedge B_{m+1} \wedge B_{m+2})} \ \mathsf{CCI}_2$$

Subcase R_{m+2}:

$$\frac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq m+2\} \quad B_1, \dots, B_{m+2} \Rightarrow D}{(A_1 > B_1), \dots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D)} \ \mathsf{R}_{m+2}$$

is deconstructed using the formula $(C \wedge \bigwedge_{i \leq m} B_i > B_{m+1} \wedge B_{m+2})$ into

$$\frac{\begin{array}{c}\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : i \leq m\} \\ C, B_1, \dots, B_m \Rightarrow C \wedge \bigwedge_{i \leq m} B_i \\ B_1, \dots, B_m, B_{m+1} \wedge B_{m+2} \Rightarrow D\end{array}}{(A_1 > B_1), \dots, (A_m > B_m), (C \wedge \bigwedge_{i \leq m} B_i > B_{m+1} \wedge B_{m+2}) \Rightarrow (C > D)} \ \mathsf{R}_{m+1}$$

$$\frac{C \wedge \bigwedge_{i \leq m} B_i \Rightarrow A_{m+1} \quad C \wedge \bigwedge_{i \leq m} B_i, B_{m+1} \Rightarrow A_{m+2} \quad B_{m+1}, B_{m+2} \Rightarrow B_{m+1} \wedge B_{m+2}}{(A_{m+1} > B_{m+1}), (A_{m+2} > B_{m+2}) \Rightarrow (C \wedge \bigwedge_{i \leq m} B_i > B_{m+1} \wedge B_{m+2})} \ \mathsf{R}_2$$

Subcase RI_{m+2}:

$$\frac{\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq m+2\} \quad B_1, \dots, B_{m+2}, C \Rightarrow D}{(A_1 > B_1), \dots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D)} \ \mathsf{RI}_{m+2}$$

is deconstructed using the formula $(C \wedge \bigwedge_{i \leq m} B_i > C \wedge \bigwedge_{i \leq m+2} B_i)$ into

$$\dfrac{\begin{array}{l}\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq m\} \\ \quad C, B_1, \dots, B_m \Rightarrow C \wedge \bigwedge_{i \leq m} B_i \\ \qquad B_1, \dots, B_m, C \wedge \bigwedge_{i \leq m+2} B_i, C \Rightarrow D\end{array}}{(A_1 > B_1), \dots, (A_m > B_m), (C \wedge \bigwedge_{i \leq m} B_i > C \wedge \bigwedge_{i \leq m+2} B_i) \Rightarrow (C > D)} \; \mathsf{RI}_{m+1}$$

$$\dfrac{\begin{array}{l}C \wedge \bigwedge_{i \leq m} B_i \Rightarrow A_{m+1} \\ \quad C \wedge \bigwedge_{i \leq m} B_i, B_{m+1} \Rightarrow A_{m+2} \\ \qquad C \wedge \bigwedge_{i \leq m} B_i, B_{m+1}, B_{m+2} \Rightarrow C \wedge \bigwedge_{i \leq m+2} B_i\end{array}}{(A_{m+1} > B_{m+1}), (A_{m+2} > B_{m+2}) \Rightarrow (C \wedge \bigwedge_{i \leq m} B_i > C \wedge \bigwedge_{i \leq m+2} B_i)} \; \mathsf{RI}_2$$

Subcase $\mathsf{ag\_CC}_{m+2}$:

$$\dfrac{\{C \Rightarrow A_i : 1 \leq i \leq m+2\} \quad \{D \Rightarrow B_i : 1 \leq i \leq m+2\} \quad B_1, \dots, B_{m+2} \Rightarrow D}{(A_1 > B_1), \dots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D)} \; \mathsf{ag\_CC}_{m+2}$$

is deconstructed using the formula $(C > (B_{m+1} \wedge B_{m+2}) \vee D)$ into

$$\dfrac{\begin{array}{l}\{C \Rightarrow A_i : 1 \leq i \leq m\} \\ \quad C \Rightarrow C \\ \qquad \{D \Rightarrow B_i : 1 \leq i \leq m\} \\ \qquad\quad D \Rightarrow (B_{m+1} \wedge B_{m+2}) \vee D \\ \qquad\qquad B_1, \dots, B_m, (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow D\end{array}}{(A_1 > B_1), \dots, (A_m > B_m), (C > (B_{m+1} \wedge B_{m+2}) \vee D) \Rightarrow (C > D)} \; \mathsf{ag\_CC}_{m+1}$$

and

$$\dfrac{\begin{array}{l}C \Rightarrow A_{m+1} \\ \quad C \Rightarrow A_{m+2} \\ \qquad (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow B_{m+1} \\ \qquad\quad (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow B_{m+2} \\ \qquad\qquad B_{m+1}, B_{m+2} \Rightarrow (B_{m+1} \wedge B_{m+2}) \vee D\end{array}}{(A_{m+1} > B_{m+1}), (A_{m+2} > B_{m+2}) \Rightarrow (C > (B_{m+1} \wedge B_{m+2}) \vee D)} \; \mathsf{ag\_CC}_2$$

Subcase $\mathsf{ag\_R}_{m+2}$:

$$\dfrac{\begin{array}{l}\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq m+2\} \\ \quad \{D \Rightarrow B_i : 1 \leq i \leq m+2\} \\ \qquad B_1, \dots, B_{m+2} \Rightarrow D\end{array}}{\Gamma, (A_1 > B_1), \dots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D), \Delta} \; \mathsf{ag\_R}_{m+2}$$

is deconstructed using the formula $(C \wedge \bigwedge_{i \leq m} B_i > (B_{m+1} \wedge B_{m+2}) \vee D)$ into

$$\dfrac{\begin{array}{l}\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq m\} \\ \quad C, B_1, \dots, B_m \Rightarrow C \wedge \bigwedge_{i \leq m} B_i \\ \qquad \{D \Rightarrow B_i : 1 \leq i \leq m\} \\ \qquad\quad D \Rightarrow (B_{m+1} \wedge B_{m+2}) \vee D \\ \qquad\qquad B_1, \dots, B_m, (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow D\end{array}}{(A_1 > B_1), \dots, (A_m > B_m), (C \wedge \bigwedge_{i \leq m} B_i > (B_{m+1} \wedge B_{m+2}) \vee D) \Rightarrow (C > D)} \; \mathsf{ag\_R}_{m+1}$$

and

$$\frac{\begin{array}{l} C \land \bigwedge_{i\leq m} B_i \Rightarrow A_{m+1} \\ \quad C \land \bigwedge_{i\leq m} B_i, B_{m+1} \Rightarrow A_{m+2} \\ \qquad (B_{m+1} \land B_{m+2}) \lor D \Rightarrow B_{m+1} \\ \qquad\quad (B_{m+1} \land B_{m+2}) \lor D \Rightarrow B_{m+2} \\ \qquad\qquad B_{m+1}, B_{m+2} \Rightarrow (B_{m+1} \land B_{m+2}) \lor D \end{array}}{(A_{m+1} > B_{m+1}), (A_{m+2} > B_{m+2}) \Rightarrow (C \land \bigwedge_{i\leq m} B_i > (B_{m+1} \land B_{m+2}) \lor D)}\ \mathsf{ag\_R}_2$$

Subcase $\mathsf{c\_ag\_CC}_{m+2}$: The construction is the same as for the subcase $\mathsf{ag\_CC}_{m+2}$, i.e.:

$$\frac{\begin{array}{l} \{C \Rightarrow A_i : 1 \leq i \leq m+2\} \\ \quad \{D \Rightarrow B_i : 1 \leq i \leq m+2\} \\ \qquad B_1, \dots, B_{m+2} \Rightarrow D \\ \qquad\quad \nvdash C, D \Rightarrow \end{array}}{(A_1 > B_1), \dots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D)}\ \mathsf{ag\_CC}_{m+2}$$

is deconstructed using the formula $(C > (B_{m+1} \land B_{m+2}) \lor D)$ into

$$\frac{\begin{array}{l} \{C \Rightarrow A_i : 1 \leq i \leq m\} \\ \quad C \Rightarrow C \\ \qquad \{D \Rightarrow B_i : 1 \leq i \leq m\} \\ \qquad\quad D \Rightarrow (B_{m+1} \land B_{m+2}) \lor D \\ \qquad\qquad B_1, \dots, B_m, (B_{m+1} \land B_{m+2}) \lor D \Rightarrow D \\ \qquad\qquad\quad \nvdash C, D \Rightarrow \end{array}}{(A_1 > B_1), \dots, (A_m > B_m), (C > (B_{m+1} \land B_{m+2}) \lor D) \Rightarrow (C > D)}\ \mathsf{c\_ag\_CC}_{m+1}$$

and

$$\frac{\begin{array}{l} C \Rightarrow A_{m+1} \\ \quad C \Rightarrow A_{m+2} \\ \qquad (B_{m+1} \land B_{m+2}) \lor D \Rightarrow B_{m+1} \\ \qquad\quad (B_{m+1} \land B_{m+2}) \lor D \Rightarrow B_{m+2} \\ \qquad\qquad B_{m+1}, B_{m+2} \Rightarrow (B_{m+1} \land B_{m+2}) \lor D \\ \qquad\qquad\quad \nvdash C, (B_{m+1} \land B_{m+2}) \lor D \Rightarrow \end{array}}{(A_{m+1} > B_{m+1}), (A_{m+2} > B_{m+2}) \Rightarrow (C > (B_{m+1} \land B_{m+2}) \lor D)}\ \mathsf{c\_ag\_CC}_2$$

Thus we also have to show that $\nvdash C, (B_{m+1} \land B_{m+2}) \lor D \Rightarrow$. Assume that $\vdash C, (B_{m+1} \land B_{m+2}) \lor D \Rightarrow$. Then by invertibility of the propositional rules we also have $\vdash C, D \Rightarrow$, in contradiction to $\nvdash C, D$. Thus the statement holds.

Subcase $\mathsf{c\_ag\_R}_{m+2}$: Again, the construction itself is the same as for the subcase $\mathsf{ag\_R}_{m+2}$, i.e.:

$$\frac{\begin{array}{ll} \{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq m+2\} & \\ \quad \{D \Rightarrow B_i : 1 \leq i \leq m+2\} & \\ \qquad B_1, \dots, B_{m+2} \Rightarrow D & \nvdash C, D \Rightarrow \end{array}}{\Gamma, (A_1 > B_1), \dots, (A_{m+2} > B_{m+2}) \Rightarrow (C > D), \Delta}\ \mathsf{c\_ag\_R}_{m+2}$$

is deconstructed using the formula $(C \wedge \bigwedge_{i \leq m} B_i > (B_{m+1} \wedge B_{m+2}) \vee D)$ into

$$\frac{\begin{array}{l}\{C, B_1, \dots, B_{i-1} \Rightarrow A_i : 1 \leq i \leq m\} \\ \quad C, B_1, \dots, B_m \Rightarrow C \wedge \bigwedge_{i \leq m} B_i \\ \qquad \{D \Rightarrow B_i : 1 \leq i \leq m\} \\ \qquad\quad D \Rightarrow (B_{m+1} \wedge B_{m+2}) \vee D \\ \qquad\qquad B_1, \dots, B_m, (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow D \\ \qquad\qquad\quad \nvdash C, D \Rightarrow \end{array}}{\Gamma, (A_1 > B_1), \dots, (A_m > B_m), (C \wedge \bigwedge_{i \leq m} B_i > (B_{m+1} \wedge B_{m+2}) \vee D) \Rightarrow (C > D), \Delta} \ \mathsf{c\_ag\_R}_{m+1}$$

$$\frac{\begin{array}{l} C \wedge \bigwedge_{i \leq m} B_i \Rightarrow A_{m+1} \\ \quad C \wedge \bigwedge_{i \leq m} B_i, B_{m+1} \Rightarrow A_{m+2} \\ \qquad (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow B_{m+1} \\ \qquad\quad (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow B_{m+2} \\ \qquad\qquad B_{m+1}, B_{m+2} \Rightarrow (B_{m+1} \wedge B_{m+2}) \vee D \\ \qquad\qquad\quad \nvdash C \wedge \bigwedge_{i \leq m} B_i, (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow \end{array}}{(A_{m+1} > B_{m+1}), (A_{m+2} > B_{m+2}) \Rightarrow (C \wedge \bigwedge_{i \leq m} B_i > (B_{m+1} \wedge B_{m+2}) \vee D)} \ \mathsf{c\_ag\_R}_2$$

Again we thus need to show that the underivability statement is true, i.e., that $\nvdash C \wedge \bigwedge_{i \leq m} B_i, (B_{m+1} \wedge B_{m+2}) \vee D \Rightarrow$. Assume otherwise. Then by invertibility of the propositional rules we also have $\vdash C \wedge \bigwedge_{i \leq m} B_i, B_{m+1} \wedge B_{m+2} \Rightarrow$ and hence $\vdash C, B_1, \dots, B_{m+2} \Rightarrow$. Together with $\vdash D \Rightarrow B_i$ for $i \leq m + 2$ and admissibility of Contraction this gives $\vdash C, D \Rightarrow$, in contradiction to $\nvdash C, D \Rightarrow$. □