Input/Output Consequence Relations: Reasoning with Intensional Contexts

Walter Carnielli[1], Marcelo E. Coniglio[1] and Leendert van der Torre[2]
[1]Centre for Logic, Epistemology and the History of Science (CLE)
and Department of Philosophy (IFCH)
State University of Campinas - UNICAMP, Brazil
[2]ICR- Computer Science and Communication
University of Luxembourg

Abstract

The input/output (I/O) logics introduced by D. Makinson and L. van der Torre in 2000 (and subsequently treated in several other papers) focus on a new kind of abstract reasoning structure devoted to the study of conditional goals and obligations. The original I/O logics dealt exclusively with the classical propositional consequence relation by adding certain "black boxes" which permit to distinguish between input and output propositions, also allowing a fine control of reusability. The present paper generalizes the input/output logics towards abstract Tarskian consequence relations, extending some perspectives initiated by Makinson and van der Torre and showing that many of the original results hold in a very general setting. This gives rise to new forms of general I/O consequence relations in the sense that each Tarskian logic has associated to it different input/output inference engines, with promising capabilities for reasoning in intensional scenarios. It is also possible to regard such I/O consequence relations as a method for combining logics, and to extend the original I/O logics to multiple-conclusion consequence versions, which would make the treatment of the linguistic operators more symmetric.

1 Introduction

In a range of contexts, one comes across processes resembling inference, but where input propositions are not in general included among outputs, and the operation is not in any way reversible. Examples arise in contexts of conditional obligations, goals, ideals, preferences, actions, and beliefs. Makinson and van der Torre's input/output logic [12] develops a theory of such input/output operations. They single out four of such operations: simple-minded, basic (making intelligent use of disjunctive inputs), simple-minded reusable (in which outputs may be recycled as inputs), and basic reusable. They are defined semantically and characterised by derivation rules, as well as in terms of relabeling procedures and modal operators. Their behaviour is studied on both semantic and syntactic levels.

Over the past decade, various extensions to their theory have been developed, inspired by applications of input/output logics in, for example, normative, multiagent and causal reasoning.

Constraints for contrary-to-duty obligations. Makinson and van der Torre [13] study what happens when input/output logics are constrained to render output consistent with input. This is of interest for deontic logic, where it provides a manner of handling contrary-to-duty obligations.

Two sets of rules for conditional permission. Makinson and van der Torre [15] and Boella and van der Torre [6] study conditional permission. This perspective provides a clear separation of the familiar notion of negative permission from the more elusive one of positive permission. Moreover, it reveals that there are at several kinds of positive permission.

Interpretations of rules for causal reasoning. Bochman [2] introduces logical formalisms of production and causal inference relations based on input/output logics. These inference relations are assigned both standard semantics (giving interpretation to their rules), and natural nonmonotonic semantics based on the principle of explanation closure. The resulting nonmonotonic formalisms provide a logical representation of abductive reasoning, and the results of the study suggest production and causal inference as general nonmonotonic formalisms providing an alternative representation for a significant part of nonmonotonic reasoning.

These and other applications lead to several questions concerning the input/output logic framework, such as the following.

Which rules? Makinson and van der Torre observe that one might consider strengthening, weakening, or otherwise modifying the systems studied in their paper, with either a purely formal motivation or an eye to possible applications. For example, with an interest in defeasible conditionals, one might drop the strengthening of the input rule, perhaps replacing it by a rule of replacement of equivalent input propositions, or in the system of simple-minded reusable output, say, one may replace cumulative transitivity by plain transitivity. In particular in the context of conditional permission, many other rules than the usual ones have been considered.

Which base logic? Some of the applications use weaker logics than propositional logic, for example logics without disjunction, see, e.g., [7]. Often more expressive logics are considered, for example first order logics, or modal logic for obligations to relate input/output logic to deontic logic [11], knowledge operators for epistemic norms, or for temporal expressions [5, 3].

How to combine logics? A single input/output logic can be seen as a way to combine two logics. For example, in Alchourrón and Bulygin's theory of normative systems [1], the input of a normative system is a set of cases, and the output is a judgment on the deontic status of the case. Moreover, input/output logics can themselves be combines, leading to logical input/output nets for architectures. Makinson and van der Torre [15] suggest to investigate structured assemblies of input/output operations. Such structures, called logical input/output nets, or lions for short, are graphs, with the nodes labelled by a normative code and an input/output operations (or recursively, by other lions). The relation of the graph indicates which nodes have access to others, providing passage for the transmission of local outputs as local inputs. The graph is further equipped with an entry point and an exit point, for global input and output.

How to allow possibly infinite sets of formulas as inputs? The input may be infinite, and when the considered logics are not compact, we need infinitary input/output logics.

How to generalize to multiple consequence? When constraints are applied to an input/output logic, there may be a set of outputs rather than a single one. Multiple consequence operations can be used to study input/output logics under constraints.

How to apply conjunction or negation to rules? The only formal framework to evaluate and classify normative system change methods is the so-called AGM frame- work of theory change. Boella, Pigozzi and van der Torre [4] therefore propose, as a normative framework for normative system change, to replace propositional formulas in the AGM framework of theory change by pairs of propositional formulas, representing the rule based character of norms, and to add several principles from the input/output logic framework. They show that some of the AGM properties cannot be expressed, because conjunction and negation of rules is not defined in the input/output logic framework.

In this paper we consider new forms of general I/O consequence relations, in the sense that each Tarskian logic has associated to it different input/output inference engines, which contributes to the open problems above. We replace propositional logic by abstract logics (which logics?) and we define input/output consequence relations, a natural setting to consider other properties of rules (which rules?). We consider non-compact (that is, infinitary) i/o logics, by allowing possibly infinite sets of formulas as inputs (infinite inputs?). We suggest how one can consider two Tarskian consequence operators instead of just one (how to combine logics?) and how to recast of (classical) I/O logics in terms of multiple-conclusion consequence relations (how to generalize to multiple consequence?).

2 Input/Output from general logics

This section is dedicated to showing how the notion of I/O logics introduced in [12] can be extended to general consequence systems. The new systems obtained offer a rich environment for reasoning in intensional scenarios involving for instance preferences, obligations, actions and beliefs, and the present paper concentrates on developing the foundational basis for such an approach. The aim here is to show how to define I/O consequence relations associated to abstract Tarskian logics and to study some of their basic properties. It is convenient first to recall the definition of abstract consequence relations attributed to A. Tarski (see [17, 16]):

Definition 2.1 A (Tarskian) consequence relation over a language L is a relation $\vdash \subseteq \wp(L) \times L$ satisfying the following properties:

(S1)	$\Gamma, \varphi \vdash \varphi$	(overlap)
(S2)	$[\Gamma \vdash \varphi$ and $(\forall \psi \in \Gamma)(\Sigma \vdash \psi)]$ implies $\Sigma \vdash \varphi$	(full cut)
(S3)	$\Gamma \vdash \varphi$ implies $\Gamma, \Sigma \vdash \varphi$	(dilution)

□

Other properties of consequence relations can be taken into consideration too; in particular,

(S4) $\Gamma \vdash \varphi$ implies $\Gamma_0 \vdash \varphi$ for some finite $\Gamma_0 \subseteq \Gamma$ (compactness)

As it is well known, a consequence relation over L is equivalent to a consequence operator $Cn : \wp(L) \to \wp(L)$ satisfying appropriate properties. Consequence relations and consequence operators are related as follows:

$$Cn(\Gamma) =_{df} \{\varphi : \Gamma \vdash \varphi\} \ (Cn \text{ from } \vdash); \qquad \Gamma \vdash \varphi \ \text{iff}_{df} \ \varphi \in Cn(\Gamma) \ (\vdash \text{ from } Cn)$$

A pair $\mathcal{L} = \langle L, \vdash \rangle$ with $\vdash$ as in Definition 2.1 is called *an abstract logic* over L. We say that a logic $\mathcal{L}$ *has conjunction and disjunction* if L has binary connectives $\wedge$ and $\vee$ satisfying the usual (classical) properties. In particular, *distributivity* holds:

$$\delta \wedge (\varphi \vee \psi) \vdash (\delta \wedge \varphi) \vee (\delta \wedge \psi) \quad \text{and} \quad (\delta \wedge \varphi) \vee (\delta \wedge \psi) \vdash \delta \wedge (\varphi \vee \psi)$$

Analogously, $\mathcal{L}$ *has an implication* if it has a conjunction $\wedge$ together with a binary connective $\to$ satisfying: $\Gamma, \varphi \wedge \psi \vdash \gamma$ iff $\Gamma, \varphi \vdash \psi \to \gamma$.

Based on classical logic, I/O logics (cf. [12]) are defined from a set of generators $G \subseteq L \times L$ such that $(\top, \top) \in G$ (where $\top$ is a given tautology), by using some of the rules below:

$$\text{(SI)} \quad \frac{(\psi,\varphi) \qquad \gamma \vdash \psi}{(\gamma,\varphi)}$$

$$\text{(WO)} \quad \frac{(\psi,\varphi) \qquad \varphi \vdash \gamma}{(\psi,\gamma)}$$

$$\text{(AND)} \quad \frac{(\psi,\varphi) \qquad (\psi,\gamma)}{(\psi,\varphi \wedge \gamma)}$$

$$\text{(OR)} \quad \frac{(\psi,\varphi) \qquad (\gamma,\varphi)}{(\psi \vee \gamma,\varphi)}$$

$$\text{(CT)} \quad \frac{(\psi,\varphi) \qquad (\psi \wedge \varphi,\gamma)}{(\psi,\gamma)}$$

Using these rules, four basic I/O logics, G_1 to G_4, can be defined:

- G_1 is generated by G by using the rules (SI), (WO) and (AND);
- G_2 is generated by G by using the rules (SI), (WO), (AND) and (OR);
- G_3 is generated by G by using the rules (SI), (WO), (AND) and (CT);
- G_4 is generated by G by using (SI), (WO), (AND), (OR) and (CT).

Guided by the above definitions it is possible to generalize I/O logics by means of substituting classical logic by any logic with conjunction and disjunction:

Definition 2.2 Let $\mathcal{L} = \langle L, \vdash \rangle$ be a compact logic and let $G \subseteq L \times L$ such that $(\top, \top) \in G$, where $\top \in L$ is such that $\emptyset \vdash \top$ holds.

(a) If $\mathcal{L}$ has a conjunction then:

- $G_1^{\mathcal{L}}$ is generated by G and $\mathcal{L}$ by using (SI), (WO) and (AND);
- $G_3^{\mathcal{L}}$ is generated by G and $\mathcal{L}$ by using (SI), (WO), (AND) and (CT).

(b) If $\mathcal{L}$ has a conjunction and a disjunction then:

- $G_2^{\mathcal{L}}$ is generated by G and $\mathcal{L}$ by using (SI), (WO), (AND) and (OR);
- $G_4^{\mathcal{L}}$ is generated by G and $\mathcal{L}$ by using (SI), (WO), (AND), (OR) and (CT).

□

Let $out_i^{\mathcal{L}}(G, \Gamma) = \{\varphi \ : \ (\psi_1 \wedge \ldots \wedge \psi_n, \varphi) \in G_i^{\mathcal{L}} \text{ for some } \{\psi_1, \ldots, \psi_n\} \subseteq \Gamma\}$ for $i = 1, \ldots, 4$, and let $G(\Gamma) = \{\varphi \ : \ (\psi, \varphi) \in G \text{ for some } \psi \in \Gamma\}$ for $\Gamma \subseteq L$. The following result generalizes the corresponding one from [12]:

Proposition 2.3 Let Cn be the consequence operator associated to $\mathcal{L}$.
(a) $out_1^{\mathcal{L}}(G,\Gamma) = Cn(G(Cn(\Gamma)))$;
(b) $out_3^{\mathcal{L}}(G,\Gamma) = \bigcap\{Cn(G(\Delta)) \ : \ \Gamma \subseteq \Delta = Cn(\Delta) \supseteq G(\Delta)\}$ whenever $\mathcal{L}$ has an implication.

The characterizations of out_2 and out_4 given in [12] cannot be obtained in the general case unless additional properties of $\mathcal{L}$ are required; in particular, for out_4 it is required that $\vdash (\varphi \to \psi) \vee \varphi$ and $\vdash \varphi \vee \neg\varphi$ must hold for a classical negation $\neg$ in $\mathcal{L}$; in other words, $\mathcal{L}$ must be an extension of classical propositional logic. Such limitations show how output relations such as out_2 and out_4 can be sensible to properties of linguistic connectives, and therefore their potential applications to intensional environments.

3 Input/Output consequence relations

This section discusses an additional generalization of I/O logics obtained by substituting the collection G of pairs (ψ,φ) by a collection $\overline{G}$ of pairs (Γ,φ) such that Γ is a set of formulas (the *input*) and φ is a formula (the *output*). Additionally, as done in the previous section, the underlying (classical) logic is substituted by an abstract logic with conjunction and disjunction.

In order to adapt the rules of I/O logics to this new context observe that, because of the rules of classical conjunction and disjunction, (SI), (AND), (WO) and (CT) can be equivalently expressed as

$$(\overline{\text{SI}}) \quad \frac{(\delta \wedge \psi, \varphi) \qquad \gamma \vdash \psi}{(\delta \wedge \gamma, \varphi)}$$

$$(\overline{\text{AND}}) \quad \frac{(\psi, \varphi) \qquad (\delta, \gamma)}{(\psi \wedge \delta, \varphi \wedge \gamma)}$$

$$(\overline{\text{OR}}) \quad \frac{(\delta \wedge \psi, \varphi) \qquad (\beta \wedge \gamma, \varphi)}{((\delta \wedge \beta) \wedge (\psi \vee \gamma), \varphi)}$$

$$(\overline{\text{CT}}) \quad \frac{(\psi, \varphi) \qquad (\delta \wedge \varphi, \gamma)}{(\psi \wedge \delta, \gamma)}$$

respectively. On the other hand, since (δ,δ) does not hold in general, the rule

$$(\overline{\text{WO}}) \quad \frac{(\psi, \varphi) \qquad \delta \wedge \varphi \vdash \gamma}{(\psi \wedge \delta, \gamma)}$$

is not valid. This motivates the following definition.

Definition 3.1 The *basic rules* for an I/O consequence relation $\Rightarrow$ are the following:

(SI$_c$) $$\frac{\Gamma, \psi \Rightarrow \varphi \qquad \Sigma \vdash \psi}{\Gamma, \Sigma \Rightarrow \varphi}$$

(WO$_c$) $$\frac{\Gamma \Rightarrow \psi \qquad \psi \vdash \varphi}{\Gamma \Rightarrow \varphi}$$

(AND$_c$) $$\frac{\Gamma \Rightarrow \varphi \qquad \Sigma \Rightarrow \psi}{\Gamma, \Sigma \Rightarrow \varphi \wedge \psi}$$

(OR$_c$) $$\frac{\Gamma, \varphi \Rightarrow \gamma \qquad \Sigma, \psi \Rightarrow \gamma}{\Gamma, \Sigma, \varphi \vee \psi \Rightarrow \gamma}$$

(CT$_c$) $$\frac{\Gamma \Rightarrow \psi \qquad \Sigma, \psi \Rightarrow \varphi}{\Gamma, \Sigma \Rightarrow \varphi}$$

□

The above rules correspond to the rules of I/O logics previously described in the context of abstract consequence relations. Thus, four basic I/O consequence relations can be generated from them.

Definition 3.2 Let $\mathcal{L} = \langle L, \vdash \rangle$ be a logic with conjunction and disjunction, and let $\overline{G} \subseteq \wp(L) \times L$ such that $(\{\top\}, \top) \in \overline{G}$ for a given $\top \in L$ such that $\emptyset \vdash \top$ holds. The basic *Input/Output consequence relations* over $\mathcal{L}$ are as follows:

- $\Rightarrow_1^{\mathcal{L}}$ is generated by $\overline{G}$ and $\mathcal{L}$ by using the rules (SI$_c$), (WO$_c$) and (AND$_c$);
- $\Rightarrow_2^{\mathcal{L}}$ is generated by $\overline{G}$ and $\mathcal{L}$ by using the rules (SI$_c$), (WO$_c$), (AND$_c$) and (OR$_c$);
- $\Rightarrow_3^{\mathcal{L}}$ is generated by $\overline{G}$ and $\mathcal{L}$ by using the rules (SI$_c$), (WO$_c$), (AND$_c$) and (CT$_c$);
- $\Rightarrow_4^{\mathcal{L}}$ is generated by $\overline{G}$ and $\mathcal{L}$ by using the rules (SI$_c$), (WO$_c$), (AND$_c$), (OR$_c$) and (CT$_c$).

□

Since rules can be iterated, the rule

$$\frac{\Gamma, \{\psi_1, \ldots, \psi_n\} \Rightarrow \varphi \qquad \{\Sigma \vdash \psi_i \ : \ 1 \leq i \leq n\}}{\Gamma, \Sigma \Rightarrow \varphi}$$

is easily derived from (SI$_c$). Analogous rules can be derived from (WO$_c$) and (CT$_c$). Compactness is not assumed, and so a stronger interaction between the involved sets can be required by considering possibly infinite sets.

Definition 3.3 The infinite version of rules (SI$_c$), (WO$_c$) and (CT$_c$) are as follows (in the rules below, Δ is assumed to be nonempty):

$$(\text{ISI}_c)\quad \frac{\Gamma, \Delta \Rightarrow \varphi \qquad \{\Sigma \vdash \psi \ : \ \psi \in \Delta\}}{\Gamma, \Sigma \Rightarrow \varphi}$$

$$(\text{IWO}_c)\quad \frac{\{\Gamma \Rightarrow \psi \ : \ \psi \in \Delta\} \qquad \Delta \vdash \varphi}{\Gamma \Rightarrow \varphi}$$

$$(\text{ICT}_c)\quad \frac{\{\Gamma \Rightarrow \psi \ : \ \psi \in \Delta\} \qquad \Sigma, \Delta \Rightarrow \varphi}{\Gamma, \Sigma \Rightarrow \varphi}$$

□

Of course each rule of Definition 3.1 can be obtained from its infinite version: it is enough to take $\Delta = \{\psi\}$. The converse is not valid however, since compactness for $\Rightarrow$ is not allowed. Thus, by combining the rules of definitions 3.1 and 3.3, several new I/O consequence relations can be generated. In particular, the following result holds.

Proposition 3.4 Let Cn be the consequence operator associated to $\mathcal{L}$, and set $\overline{G}(\Gamma) = \{\psi \ : \ (\Sigma, \psi) \in \overline{G} \ \text{ for some } \Sigma \subseteq \Gamma\}$. Let $\Rightarrow_1^{I,\mathcal{L}}$ be the I/O consequence relation generated by $\overline{G}$ and $\mathcal{L}$ by using the rules (ISI_c), (IWO_c) and (AND_c), and let $out_1^{I,\mathcal{L}}(\overline{G}, \Gamma) = \{\varphi \ : \ \Gamma \Rightarrow_1^{I,\mathcal{L}} \varphi\}$. Then $out_1^{I,\mathcal{L}}(\overline{G}, \Gamma) = Cn(\overline{G}(Cn(\Gamma)))$.

Proof: Suppose that $\varphi \in Cn(\overline{G}(Cn(\Gamma)))$. Then $\Delta \vdash \varphi$ for $\Delta = \overline{G}(Cn(\Gamma))$. Let $\psi \in \Delta$; then there is Σ such that $(\Sigma, \psi) \in \overline{G}$ and $\Gamma \vdash \delta$ for every $\delta \in \Sigma$. Thus $\Gamma \Rightarrow_1^{Is,\mathcal{L}} \psi$ follows by (ISI_c), for every $\psi \in \Delta$. Then, by (IWO_c) it follows that $\Gamma \Rightarrow_1^{I,\mathcal{L}} \varphi$, that is, $\varphi \in out_1^{I,\mathcal{L}}(\overline{G}, \Gamma)$.

Conversely, assume that $\varphi \in out_1^{I,\mathcal{L}}(\overline{G}, \Gamma)$. By induction on the length n of a $\Rightarrow_1^{I,\mathcal{L}}$-derivation of φ from Γ we will prove that $\overline{G}(Cn(\Gamma)) \vdash \varphi$. If $n = 1$ then $(\Gamma, \varphi) \in \overline{G}$ and so $\varphi \in \overline{G}(Cn(\Gamma))$, because $\Gamma \subseteq Cn(\Gamma)$; thus $\overline{G}(Cn(\Gamma)) \vdash \varphi$. Suppose the result holds for any derivation of length $k \leq n$, and suppose that $\Gamma \Rightarrow_1^{I,\mathcal{L}} \varphi$ is derived in $n + 1$ steps. If $(\Gamma, \varphi) \in \overline{G}$ then the proof is as above. If $\Gamma, \Sigma \Rightarrow_1^{I,\mathcal{L}} \varphi$ follows from $\Gamma, \Delta \Rightarrow_1^{I,\mathcal{L}} \varphi$ and $\{\Sigma \vdash \psi \ : \ \psi \in \Delta\}$ by (ISI_c) then $\overline{G}(Cn(\Gamma \cup \Delta)) \vdash \varphi$, by induction hypothesis. By $\{\Sigma \vdash \psi \ : \ \psi \in \Delta\}$ and full cut it follows that $Cn(\Gamma \cup \Delta) \subseteq Cn(\Gamma \cup \Sigma)$ and so $\overline{G}(Cn(\Gamma \cup \Delta)) \subseteq \overline{G}(Cn(\Gamma \cup \Sigma))$, therefore $\overline{G}(Cn(\Gamma \cup \Sigma)) \vdash \varphi$. If $\Gamma \Rightarrow_1^{I,\mathcal{L}} \varphi$ follows from $\{\Gamma \Rightarrow_1^{I,\mathcal{L}} \psi \ : \ \psi \in \Delta\}$ and $\Delta \vdash \varphi$ by (IWO_c) then $\overline{G}(Cn(\Gamma)) \vdash \psi$ for every $\psi \in \Delta$, by induction hypothesis; since $\Delta \vdash \varphi$ then $\overline{G}(Cn(\Gamma)) \vdash \varphi$, by full cut. Finally, if $\Gamma, \Sigma \Rightarrow_1^{I,\mathcal{L}} \varphi \wedge \psi$ follows from $\Gamma \Rightarrow_1^{I,\mathcal{L}} \varphi$ and $\Sigma \Rightarrow_1^{I,\mathcal{L}} \psi$ by (AND_c) then, by induction hypothesis, $\overline{G}(Cn(\Gamma)) \vdash \varphi$ and $\overline{G}(Cn(\Sigma)) \vdash \psi$. Since $\overline{G}(Cn(\Gamma)) \cup \overline{G}(Cn(\Sigma)) \subseteq \overline{G}(Cn(\Gamma \cup \Sigma))$ then $\overline{G}(Cn(\Gamma \cup \Sigma)) \vdash \varphi$ and $\overline{G}(Cn(\Gamma \cup \Sigma)) \vdash \psi$, therefore $\overline{G}(Cn(\Gamma \cup \Sigma)) \vdash \varphi \wedge \psi$.
■

4 Further perspectives

The previous results warrant the basic foundational framework for the extensions of I/O logics towards abstract Tarskian consequence relations. This does not exhaust the field, however, as further horizons and perspectives for I/O consequence relations can be envisaged, as briefly outlined here.

Recall the rules of definitions 3.1 and 3.3. Observe that, in (SI_c), the consequence relation $\vdash$ is applied to inputs, whereas $\vdash$ is applied to outputs in rule (WO_c); analogous remarks applies to (ISI_c) and (WO_c), respectively. Suppose now that inputs and outputs are defined over the same language L, but inputs are governed by an internal logic $\mathcal{L}_1 = \langle L, \vdash_1 \rangle$ with disjunction whereas the underlying logic for outputs, $\mathcal{L}_2 = \langle L, \vdash_2 \rangle$, has a conjunction. Then, $\vdash$ can be substituted by $\vdash_2$ in (SI_c) and (ISI_c); on the other hand, $\vdash$ can be substituted by $\vdash_1$ in (WO_c) and (IWO_c). Thus, given a set of generators $\overline{G} \subseteq \wp(L) \times L$, an input-output consequence relation defined by means of the new rules (while keeping unchanged the other rules) can be regarded as a method for defining a combination of logics $\mathcal{L}_1$ and $\mathcal{L}_2$. Moreover, the set of generators could be seen as a set of bridge principles, or even a generalized translation from $\mathcal{L}_1$ to $\mathcal{L}_2$ relating the logic of inputs and the logic of outputs. Universal properties of the combined reasoning could be analyzed, along the same lines of algebraic fibring (see, for instance, [9]).

Other interesting generalization of I/O consequence relations could be obtained by departing from logics defined by multiple-conclusions consequence relations instead of single-conclusion ones, as done in Section 3. Recall (cf. [16]) that a quasi-partition of a set Σ is a pair of sets (Θ, Θ') such that $\Theta \cup \Theta' = \Sigma$ and $\Theta \cap \Theta' = \emptyset$. By denoting by $QP(\Sigma)$ the set of quasi-partitions of Σ, a multiple-conclusion consequence relation over a language L is a relation $\vdash \subseteq \wp(L) \times \wp(L)$ satisfying the following properties:

(M1) $\Gamma, \varphi \vdash \Delta, \varphi$

(M2) $[(\exists \Sigma \subseteq L)(\forall (\Theta, \Theta') \in QP(\Sigma)(\Gamma, \Theta \vdash \Delta, \Theta')]$ implies $\Gamma \vdash \Delta$

(M3) $\Gamma \vdash \Delta$ implies $\Gamma, \Sigma \vdash \Delta, \Theta$

The adaptation of the rules introduced in Section 3 to the multiple-conclusion scenario is an interesting challenge.

Finally, the model-theoretic framework of *transfers* (cf. [10]) for studying propositional logics and their properties seems particularly well-suited for representing I/O consequence relations. In particular, the notion of *morphism* (or transfer) between two I/O consequence relations can be naturally defined within this framework.

Acknowledgements: The first two authors acknowledge support from The State of São Paulo Research Foundation (FAPESP), Brazil, Thematic Project number 2004/14107-2 ("ConsRel"), and by individual research grants from The National Council for Scientific and Technological Development (CNPq), Brazil. Carnielli also acknowledges support from the Fonds National de la Recherche Luxembourg for a sabbatical stay at the ICR- University of Luxembourg.

References

[1] C.E. Alchourrón and E. Bulygin (1971) *Normative Systems.* Springer Verlag.

[2] A. Bochman (2004) A causal approach to nonmonotonic reasoning. *Artif. Intell.* 160(1-2): 105-143

[3] G. Boella, J. Broersen, and L. van der Torre: Reasoning about Constitutive Norms, Counts-As Conditionals, Institutions, Deadlines and Violations. In *Intelligent Agents and Multi-Agent Systems, 11th Pacific Rim International Conference on Multi-Agents, PRIMA 2008*, Hanoi, Vietnam, December 15-16, 2008. Proceedings. Lecture Notes in Computer Science 5357 Springer 2008: 86-97.

[4] G. Boella, G. Pigozzi, and L. van der Torre (2009) Normative Framework for Normative System Change. In *Proceedings of AAMAS 2009.*

[5] G. Boella, J. Hulstijn, and L. van der Torre (2005) Interaction in Normative Multi-Agent Systems. *Electr. Notes Theor. Comput. Sci.* 141(5): 135-162

[6] G. Boella and L. van der Torre (2008) Institutions with a hierarchy of authorities in distributed dynamic environments. *Artif. Intell. Law* 16(1): 53-71.

[7] G. Boella and L. van der Torre (2008): Substantive and procedural norms in normative multiagent systems. *J. Applied Logic* 6(2): 152-171.

[8] D.J. Brown and R. Suszko (1973) Abstract Logics. *Dissertationes Mathematicae*, v. 102, p. 9–41.

[9] W.A. Carnielli, M.E. Coniglio, D. Gabbay, P. Gouveia and C. Sernadas. *Analysis and Synthesis of Logics. How to Cut and Paste Reasoning Systems.* Volume 35 in the Applied Logic Series, Springer, 2008.

[10] M.E. Coniglio and W.A. Carnielli (2002) Transfers between logics and their applications. *Studia Logica*, v. 72, n. 3, p. 367–400.

[11] J. Hansen, G. Pigozzi and L. van der Torre (2007) Ten Philosophical Problems in Deontic Logic. In G. Boella, L. van der Torre and H. Verhagen (Eds.): *Normative Multi-agent Systems*, 18.03. - 23.03.2007. Dagstuhl Seminar Proceedings 07122 Internationales Begegnungs- und Forschungszentrum fr Informatik (IBFI), Schloss Dagstuhl, Germany 2007

[12] D. Makinson and L. van der Torre (2000) Input/Output Logics. *Journal of Philosophical Logic* , v. 29, p. 383–408.

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

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

[15] D. Makinson and L. van der Torre (2003), What is Input/Output Logic? In *Foundations of the Formal Sciences II: Applications of Mathematical Logic in Philosophy and Linguistics*. Trends in Logic, Volume 17, Kluwer, 2003. Extended version appeared in G. Boella, L. van der Torre and H. Verhagen (Eds.): *Normative Multi-agent Systems*, 18.03. - 23.03.2007. Dagstuhl Seminar Proceedings 07122 Internationales Begegnungs- und Forschungszentrum fr Informatik (IBFI), Schloss Dagstuhl, Germany 2007

[16] D. J. Shoesmith and T. J. Smiley (1978) *Multiple-Conclusion Logic*. Cambridge: Cambridge University Press.

[17] R. Wójcicki (1988) *Theory of logical calculi:* basic theory of consequence operations. Dordrecht: Kluwer Academic Press. (Synthese Library, v. 199)