Constrained I/O logics
Handout

X. Parent & L. van der Torre
University of Luxembourg

November 14, 2015

1 Introduction

Constrained I/O logic ("cIOL", for short) aims at giving a finer grained analysis of the notion of obligation than unconstrained I/O logic does.

The following two problems have led to the use of constraints in input/output logics:

- the question of how to deal with violations and obligations resulting from violations, known as contrary-to-duty (CTD) reasoning. It has been discussed in the context of the notorious contrary-to-duty paradoxes such as Chisholm [1]'s and Forrester [2]'s paradox;
- the question of how to accommodate deontic dilemmas (unsolvable conflicts between obligations), and the question of how to reason about conflicting obligations of different strengths.

The use of constraints has been introduced in [6] in relation to CTDs, and has been extended to the topic of conflicts in [7].

2 Norm violation

The following two examples show why contrary-to-duty scenarios create a problem in I/O logic without constraints.

Example 2.1 (Chisholm [1])**.** Let $out = out_i$, where $i \in \{3, 4\}$. Assume $N = \{(\top, h), (h, t), (\neg h, \neg t)\}$, where h and t are for *helping* and *telling*. Put $A = \{\neg h\}$. We have $out(N, A) = Cn(t, \neg t) = \mathbb{L}$, where $\mathbb{L}$ is the set of all the formulas.

Example 2.2 (Forrester [2])**.** Let $out = out_i$, where $i \in \{1, 2, 3, 4\}$. Assume $N = \{(\top, \neg k), (k, k \wedge g)\}$, where k and $k \wedge g$ are for *killing* and *killing-gently*. Put $A = \{k\}$. We have $out(N, A) = \mathbb{L}$.

The strategy is to adapt a technique that is well known in the logic of belief change—cut back the set of norms to just below the threshold of yielding excess. This amounts to carrying out a contraction on the set N of norms.

The above strategy is implemented using a set C (so-called constraints) as an extra parameter. It is used to filter out excessive output. What we get is (as Makinson calls it) the "net" output (as opposed to the gross output).

Definition 1 (Maxfamily)**.**

- *maxfamily*(N, A, C) *is the set of* $\subseteq$*-maximal subsets* N' *of* N *such that out*(N', A) *is consistent with* C.
- *outfamily*$(N, A, C) =$ $\{$*out*$(N', A) \mid N' \in$ *maxfamily*$(N, A, C)\}$.

For CTDs, it is assumed that $C = A$ (I/O constraint). The input represents something that is unalterably true, viz. that has happened and cannot be changed (cf. Hansson [4, §13]'s interpretation of circumstances). The agent has to ask himself what obligations (output) this input gives rise to: even if the input should have not come true, one has to "make the best out of the sad circumstances".

A set of norms and an input do not have a set of formulas as output, but a set of sets of formulas. We can infer a set of formulas by taking the join (credulous) or meet (skeptical).

Definition 2 (Constrained, net output)**.** *Define*

$$out_c(N, A) = out_{\cup/\cap}(N, A) = \cup/\cap \; outfamily(N, A, C)$$

Example 2.3 (Chisholm, cont'd)**.** Let out, N and A be as in example 2.1. Put $C = A$. We have *maxfamily*$(N, A, C) = \{\{(h, t), (\neg h, \neg t)\}\}$, and so $out_c(N, A) = Cn(\neg t)$.

Let $A = \{h\} = C$. Hence *maxfamily*$(N, A, C) = \{\{(\top, h), (h, t), (\neg h, \neg t)\}\}$, and so $out_c(N, A) = Cn(h, t)$.

Example 2.4 (Forrester, cont'd)**.** Let out, N and A be as in example 2.2. Put $C = A$. We have *maxfamily*$(N, A, C) = \{\{(k, k \wedge g)\}\}$, and so $out_c(N, A) = Cn(k \wedge g)$.

Let $A = \{\neg k\} = C$. Hence *maxfamily*$(N, A, C) = \{\{(\top, \neg k), (k, k \wedge g)\}\}$, and so $out_c(N, A) = Cn(\neg k)$.

Example 2.5 (Multiple levels of violation)**.** Assume $out = out_i$, where $i \in \{3,4\}$. Assume $N = \{(\top, k), (\neg k, a), (\neg k \wedge \neg a, s)\}$, where k, a and s are for *keeping a promise*, *apologising* and *being ashamed*, respectively. We have:

$$out_c(N, \neg k) = Cn(a) \quad (1)$$

$$out_c(N, \neg k \wedge \neg a) = Cn(s) \quad (2)$$

EXERCISE

Exercise 2.1. There are two main desiderata for a logic dealing with CTDs:

- the logic should give a consistent representation to CTDs scenarios
- the logic should give a representation to the sentences involved in such a way that each formula remains logically independent of the others.

Explain in what ways cIOL meets the second requirement.

3 Accommodating dilemmas

Example 3.1 illustrates why dilemmas (unsolvable conflicts between obligations) create a problem in I/O logic without constraints.

Example 3.1 (Unary conflict)**.** Let $out = out_i$, where $i \in \{1,2,3,4\}$. Assume $N = \{(a,b), (a,\neg b)\}$ and $A = \{a\}$. We have $out(N, A) = Cn(b, \neg b) = \mathbb{L}$.

To accommodate dilemmas, we use the same strategy as for CTDs. For simplicity's sake, we assume that the final output is obtained by taking the meet (skeptical).

Example 3.2 (Unary conflict, cont'd)**.** Let out, N and $A = \{a\}$ be as in Example 3.1. Put $C = \emptyset$. The maxfamily has two elements, $\{(a,b)\}$ and $\{(a,\neg b)\}$. The outfamily has two elements, $Cn(b)$ and $Cn(\neg b)$. So $out_c(N, A) = Cn(\emptyset)$.

Example 3.3 (Binary conflict)**.** Let $out = out_i$, where $i \in \{1,2,3,4\}$. Assume $N = \{(a,b), (a,c)\}$, $A = \{a\}$ and $C = \{b \to \neg c\}$. We have $out_c(N, A) = Cn(b \vee c)$.

Goble [3] identifies three main desiderata for a logic admitting the possibility of normative conflicts. They are phrased below in the modal logic notation.

Desid. 1 Make conflicts logically consistent

$$\bigcirc A, \bigcirc\neg A \nvdash \bot$$

Desid. 2 Avoid deontic explosion

$$\bigcirc A, \bigcirc\neg A \nvdash \bigcirc B$$

Desid. 3 Do not give away too much, that is keep, e.g.,

$$\bigcirc(A \vee B), \bigcirc\neg A \vdash \bigcirc B$$

The I/O approach meets these desiderata. This is illustrated below with the example of Desid. 3.

Example 3.4 (Alternative service)**.** Let out be an I/O operation. Assume $N = \{(\top, f \vee s), (\top, \neg f)\}$, where f and s are for *fighting in the army* and *performing an alternative national service*, respectively. Put $A = \{\top\}$ and $C = \emptyset$. We have $out_c(N, A) = Cn(s, \neg f)$.

Link with default-assumption consequence

There are connections between cIOL and some well-known systems for nonmonotonic reasoning developed in AI. In particular so-called Poole systems (introduced under the name "default-assumption consequence" in the *Bridges* handout) may be seen as a special case of constrained input/output logic. We recall the basic idea underpinning Poole systems: when an inconsistency is generated one looks at what follows from all the maximal consistent subsets of the set of K of background assumptions.

Let (K, A, C) be a Poole system. K, A and C are sets of formulas in the language of classical propositional logic. K is a set of background assumptions. A is the input. C is a set of constraints. (The *Bridges* handout focuses on the particular case of a Poole system without constraints, when $C = \emptyset$.)

Definition 3 (Extfamily)**.** *Given any Poole system (K, A, C), let extfamily(K, A, C) be the family of its extensions in the sense of Poole. That is,*

- *extfamily(N, A, C) is the set of $Cn(A \cup K')$, where K' is maximal among the subsets K" of K such that $A \cup K" \cup C$ is consistent.*

Theorem 1. *Given any Poole system (K, A, C), extfamily(K, A, C) = outfamily(N, A, C), where $N = \{(\top, x) : x \in K\}$, and outfamily is defined using reusable basic throughput out_4^+.*

Proof. This is [6, Obs. 4]. Hint: remember that $out_4^+(N, A) = Cn(A \cup m(N))$. □

EXERCISE

Exercise 3.1. Show that $b \in out_\cap(N, \top)$ whenever $b \vee c \in out_\cap(N, \top)$ and $\neg c \in out_\cap(N, \top)$.

4 Obligations of different strengths

Basic idea Start with a priority relation $\geq$ on norms. Lift it to a priority relation $\geq^s$ on sets of norms. Use $\geq^s$ to select a "preferred" element in the maxfamily. Restrict the final, net output to this preferred element.

Let $\geq \subseteq N \times N$. $(a, x) \geq (b, y)$ is read: (a, x) is at least as strong as (b, y). $\geq$ is required to be reflexive and transitive. (a, x) and (b, y) are said to be incomparable under $\geq$ if $(a, x) \ngeq (b, y)$ and $(b, y) \ngeq (a, x)$.

$>$ is the strict order induced by $\geq$. $(a, x) > (b, y)$ is read: (a, x) is strictly stronger than (b, y). $>$ is defined by putting $(a, x) > (b, y)$ whenever $(a, x) \geq (b, y)$ and $(b, y) \ngeq (a, x)$.

Different notions of lifting have been considered in the literature. The one defined below is named after Brass.

Definition 4 (Brass lifting)**.** $N_1 \geq^s N_2$ *if and only if* $\forall (a, x) \in N_2 - N_1 \; \exists (b, y) \in N_1 - N_2$ *such that* $(b, y) \geq (a, x)$.

The definition of $>^s$ in terms of $\geq^s$ parallels that of $>$ in terms of $\geq$.

The Brass lifting is to be contrasted with the $\forall\forall$ lifting, obtained by putting $N_1 \geq^s N_2$ iff $\forall (a, x) \in N_1 \; \forall (b, y) \in N_2 \; (a, x) \geq (b, y)$. Adapted from Goble [3], the following example explains why the Brass lifting is preferred over the $\forall\forall$ lifting.

Example 4.1 (Mission)**.** Let $(\top, a) > (\top, b) > (\top, c)$, where a, b and c are three cities my boss sends me to. Put $C = \{a \to (b \to \neg c), a \to \neg b\}$ and $A = \{\top\}$. The maxfamily has two elements, $N_1 = \{(\top, a), (\top, c)\}$ and $N_2 = \{(\top, b), (\top, c)\}$. The $\forall\forall$ definition yields that N_1 and N_2 are not comparable under $\geq^s$, while the Brass definition yields that $N_1 >^s N_2$. The second solution seems more intuitive.

Definition 5 (Preferred output, out_P)**.** *We define the preffamily*(N, A, C) *as the set of* $\geq^s$*-maximal elements of maxfamily*(N, A, C)*. We put* $out_P(N, A) = \cap / \cup$ *out*(N, A)*, where* $N \in$ *preffamily*(N, A, C)*.*

Note that in typical examples there is only one element in the preffamily.

Not-triggered high-ranking obligations raise special issues as illustrated below. Examples 4.2 and 4.3 describe a case where the least important obligation overrules a more important one. This, in order to avoid the violation of an even more important obligation.

Example 4.2 (Cancer, [7])**.** Assume $out = out_i$, where $i \in \{3, 4\}$. Let $(b, c) > (a, b) > (a, \neg b)$, where a, b and c denote a set of data, the fact of having chemo, and the fact of keeping the WBC count high enough. Put $A = C = \{a\}$. The maxfamily has two elements, $N_1 = \{(b, c), (a, \neg b)\}$ and $N_2 = \{(b, c), (a, b)\}$. The preffamily has one element, N_2. And so $out_P(N, A) = Cn(b, c)$.

Example 4.3 (Cancer, cont'd, [7])**.** Let the norms and priorities involved be as in Example 4.2. But put $A = C = \{a, \neg c\}$. The maxfamily has two elements, $N_1 = \{(b, c), (a, \neg b)\}$ and $N_2 = \{(a, b)\}$. The preffamily has one element, N_1. And so $out_P(N, A) = Cn(\neg b)$. This tallies with our intuitions: usually physicians postpone chemo. Note that most approaches from literature output b instead.

The order puzzle has been introduced by Horty in [5].

Example 4.4 (Order puzzle, [5])**.** Let $(a, \neg b) > (\top, b) > (\top, a)$, where a and b are for *putting the heating on* and *opening the window* respectively. The norms $(\top, a)$, $(\top, b)$ and $(a, \neg b)$ are issued by a priest, a bishop and a cardinal, respectively. Assume $out = out_i$, where $i \in \{3, 4\}$. Put $A = \{\top\}$ and $C = \emptyset$. The maxfamily has three elements, $N_1 = \{(\top, a), (\top, b)\}$ and $N_2 = \{(\top, a), (a, \neg b)\}$ and $N_3 = \{(\top, b), (a, \neg b)\}$. We have $N_3 >^s N_2 >^s N_1$. So the preffamily has one element, N_3. And so $out_P(N, A) = Cn(b)$.

EXERCISES

Exercise 4.1. What are the properties of $>$?

Exercise 4.2. State in full the definition of $>^s$.

References

[1] R.M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–66, 1963.

[2] J. Forrester. Gentle murder, or the adverbial samaritan. *Journal of Philosophy*, 81:193–196, 1984.

[3] L. Goble. Prima facie norms, normative conflicts and dilemmas. In D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*, page 241352. College Publications, London, 2013.

[4] B. Hansson. An analysis of some deontic logics. *Noûs*, 3(4):373–398, 1969.

[5] J. Horty. *Reasons as Defaults*. O.U.P, U.S.A., 2012.

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

[7] X. Parent. Moral particularism in the light of deontic logic. *Artif. Intell. Law*, 19(2-3):75–98, 2011.